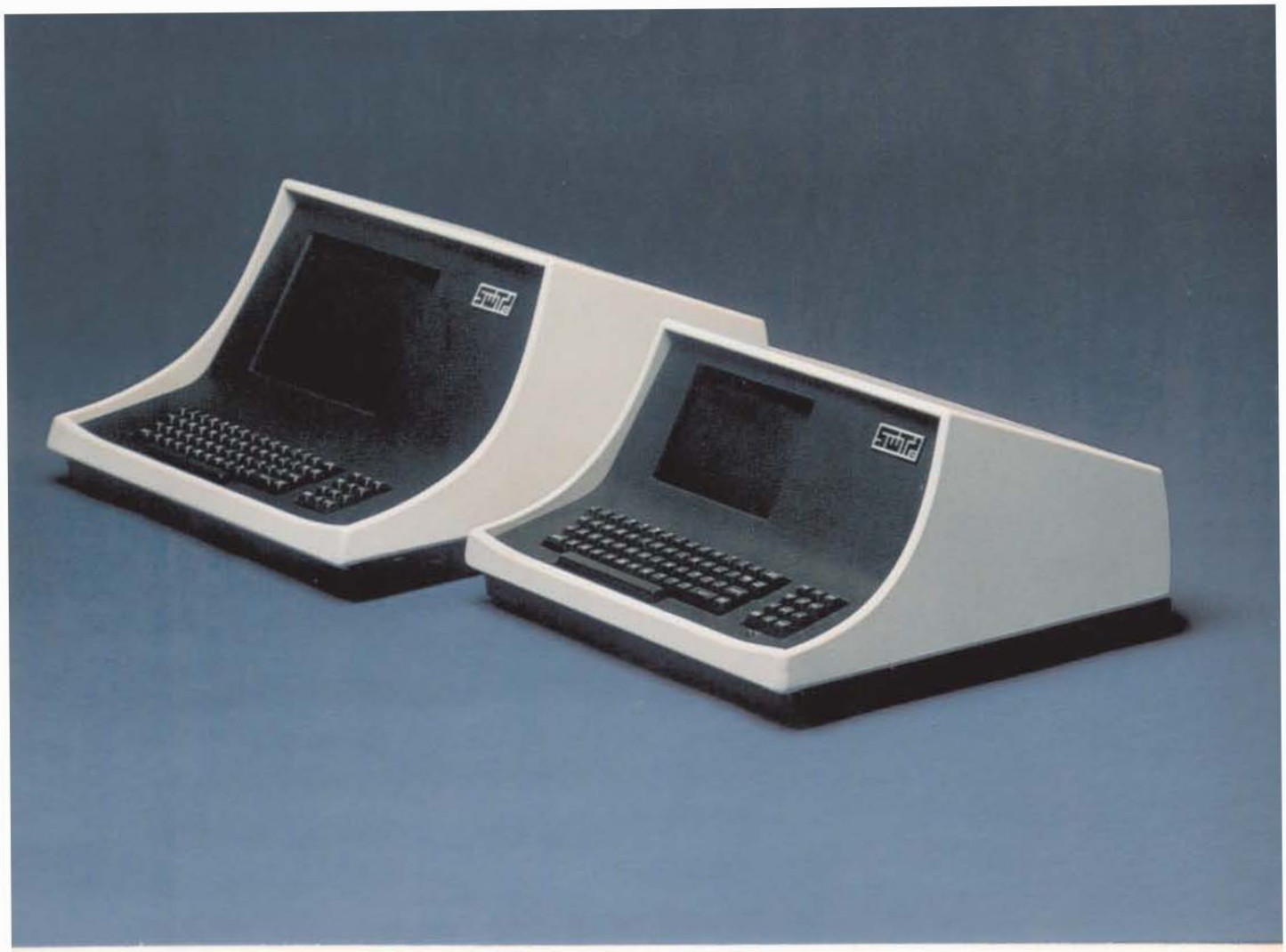
'68' MICRO JOURNAL

$2.95USA

Australia	A $ 4.75	New Zealand	NZ $ 4.7
Singapore	S $ 9.45	Hong Kong	H $23.5
Malaysia	M $ 9.45	Sweden	30:-SE

VOLUME V ISSUE VI • Devoted to the 68XX User • June 1983

"Small Computers Doing Big Things"



YOUR CHOICE-smart either way

- Over 140 software driven functions
- 82 x 24 or 82 x 20 screen format — software selectable
- High resolution 7 x 12 matrix characters — P-31 green phosphor
- Upper/lower case character set — plus graphics character set
- 56-key alphanumeric keyboard — plus 12-key cursor, numeric pad
- Internal editing functions — insert, delete, scroll, roll, slide, etc.
- Parallel printer I/O port
- 50 to 38,400 baud operation — programmable
- Cursor type, cursor position, print control characters, protected fields, shift inversion, dual intensity and many other features

8212 — twelve-inch diagnonal screen or 8209 — nine-inch diagnonal screen

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

The UniFLEX™ Operating System extracts every last drop

from the 8 bit 6809 microprocessor allowing it to outperform many 16 bit systems

With the UniFLEX™ Operating System, the 8 bit 6809 microprocessor can perform as well as larger CPUs in a multi-user, multi-tasking environment.

Independently developed from the ground up, UniFLEX™ closely models the features found in the UNIX™ Operating System. And in two years of use, UniFLEX™ has proven the abilities of the 6809 to perform large system functions when incarporated intoa propertydesigned mainframe.

Some of the features supported include:
- full multi-user, multi-tasking capabilities
- hierarchical file systems
- device independent I/O
- four Gigabyte disk capacities
- full file protection
- inter-task communication via pipes
- I/O redirection
- task swapping for efficient memory usage
- full random-access files
- comprehensive shell command language
- foreground-background jobs
- electronic mail and printer spooling
- system accounting facilities

The support software currentlyavailable for use under UniFLEX™ is extensive. A sampling of the programs available includes:
- native C compiler (full implementation)
- native Pascal compiler
- FORTRAN 77 ANSI Subset compiler
- COBOL compiler with ISAM files, Report Writer & Sort/Merge
- Extended BASIC interpreter
- Extended BASIC precompiler
- text editing and processing software
- enhanced printer spooler
- varietyofabsolute and relocatable assemblers
- debug and diagnostic packages

Technical Systems Consultants, Inc. also offers a line of single user FLEX™ software products for 6800 and 6809 processors. For those having an absolute need for a 16 bit processor, UniFLEX™ will be available through OEM licensing arrangements for the 68000 microprocessor. Please call or write for additional information on individual products or OEM licensing arrangements.

UNIX™ is a trademark of Bell Laboratories.
FLEX™ and UniFLEX™ are trademarks of Technical Systems Consultants, Inc.

technical systems consultants, inc.

111 Providence Road
Chapel Hill, North Carolina 27514
(919) 493-1451

'68' MICRO JOURNAL

Portions of the text for 68 MICRO JOURNAL was prepared using the following furnished hard/software.

COMPUTERS-HARDWARE
Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216
S 9-5/8 DMF disk-CDS1-8212W-Sprint 3 Printer

GIMIX Inc.
1337 West 37th Place
Chicago, IL 60609
Super Mainframe-OS9-FLEX-Assorted Hardware

EDITORS-WORD PROCESSORS
Technical Systems Consultants, Inc.
111 Providence Road
Chapel Hill, NC 27514
FLEX-Editor-Processor

Great Plains Computer Company, Inc.
PO Box 916
Idaho Falls, ID 83401
STYLO-Mail Merge

CONTENTS

Send All Correspondence To:

Computer Publishing Center
68 MICRO JOURNAL
5900 Cassandra Smith
PO Box 849
Hixson, TN 37343
615 842-4600

68' Micro Journal is published 12 times a year by Computer Publishing Inc. Second Class Postage Paid ISSN 0194-5025 at Hixson, Tenn. and additional entries. Postmaster: send Form 3579 to 68' Micro Journal, PO Box 849, Hixson, Tennessee.

SUBSCRIPTION RATES
USA
1-Year $24.50 2-Years $42.50 3-Years $64.50
FOREIGN
See Page 52

Items Submitted for Publication

Articles submitted for publication should be accompanied by the authors **full name, address, date and telephone number.** It is preferred that articles be submitted on either 5 or 8 inch diskette in TSC Editor format or STYLO format. All diskettes will be returned.

The following TSC Text Processor commands **ONLY** should be used (due to our proportional processor): .sp space, .pp paragraph, .fi fill and .nf no fill. Also please do not format within the text with multiple spaces. The rest we will enter at time of editing.

STYLO commands are all acceptable except the .pg page command, we print edited text files in continous text.

All articles submitted on diskettes should be in TSC FLEX™ format, either FLEX2 6800, or FLEX9 6809 any version.

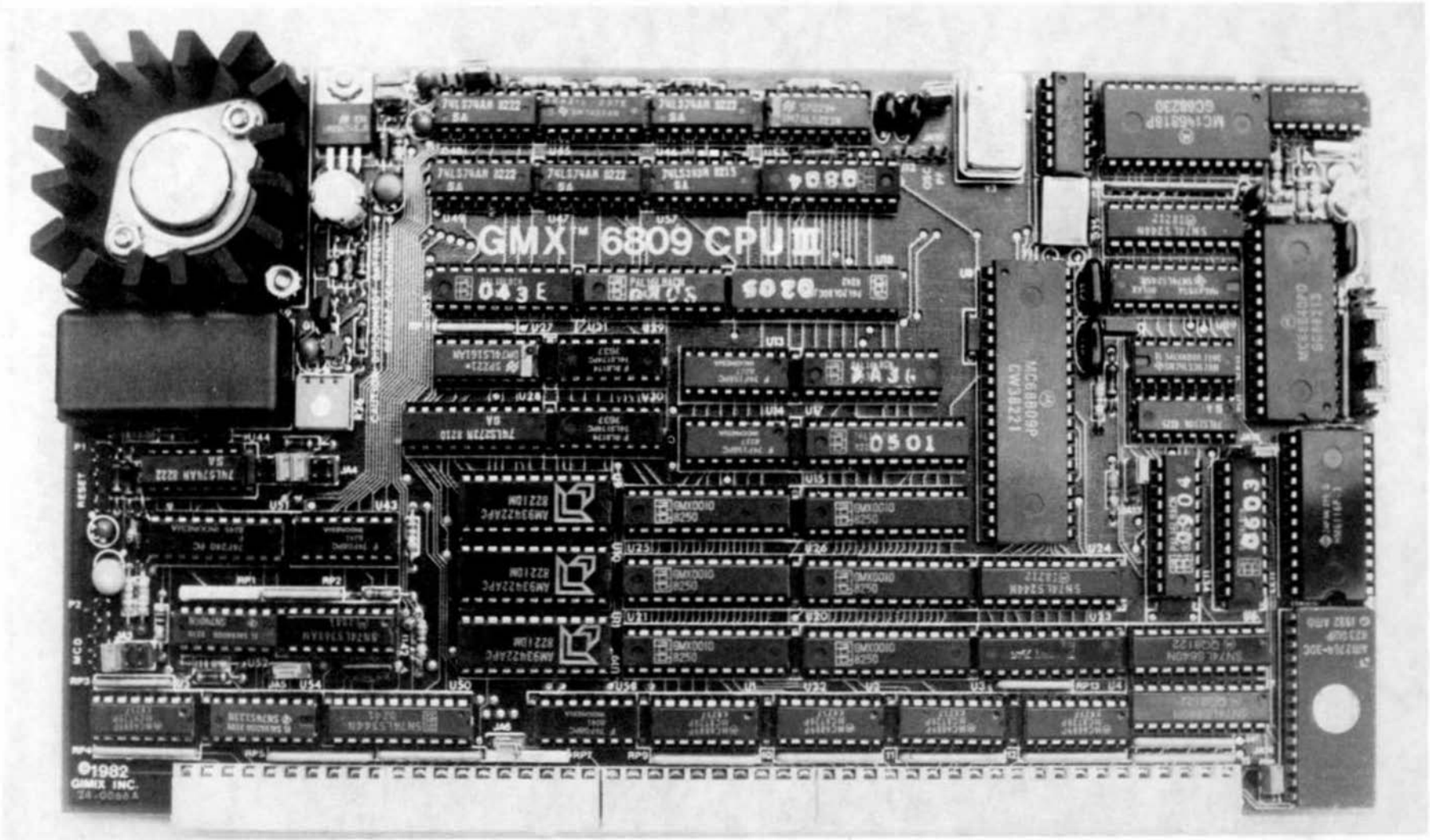
If articles are submitted on paper they should be on white 8X11 bond or better grade paper. No hand written articles (hand written or drawn art accepted). All paper submitted articles will be photo reproduced. This requires that they be typed or produced with a dark ribbon (no blue), single spaced and type font no smaller than 'elite' or 12 pitch. Typed text should be approximately 7 inches wide (will be reduced to column width of 3 1/2 inches). **Please use a dark ribbon!**

All letters to the editor should also comply with the above and bear a signature. Letters of 'gripes' as well as 'praise' are solicited. We attempt to publish all letters to the editor verbatim, however, we reserve the right to reject any submission for lack of 'good taste'. We reserve the right to define what constitutes 'good taste'.

Advertising: Commercial advertisers please contact the 68 Micro Journal advertising department for current rate sheet and requirements.

Classified: All classified must be non-commercial. Maximum 20 words per classified ad. Those consisting of more than 20 words should be figured at .35 cents per word. 20 words or less $7.50 minimum, one time, paid in advance. No classified ads accepted by telephone.

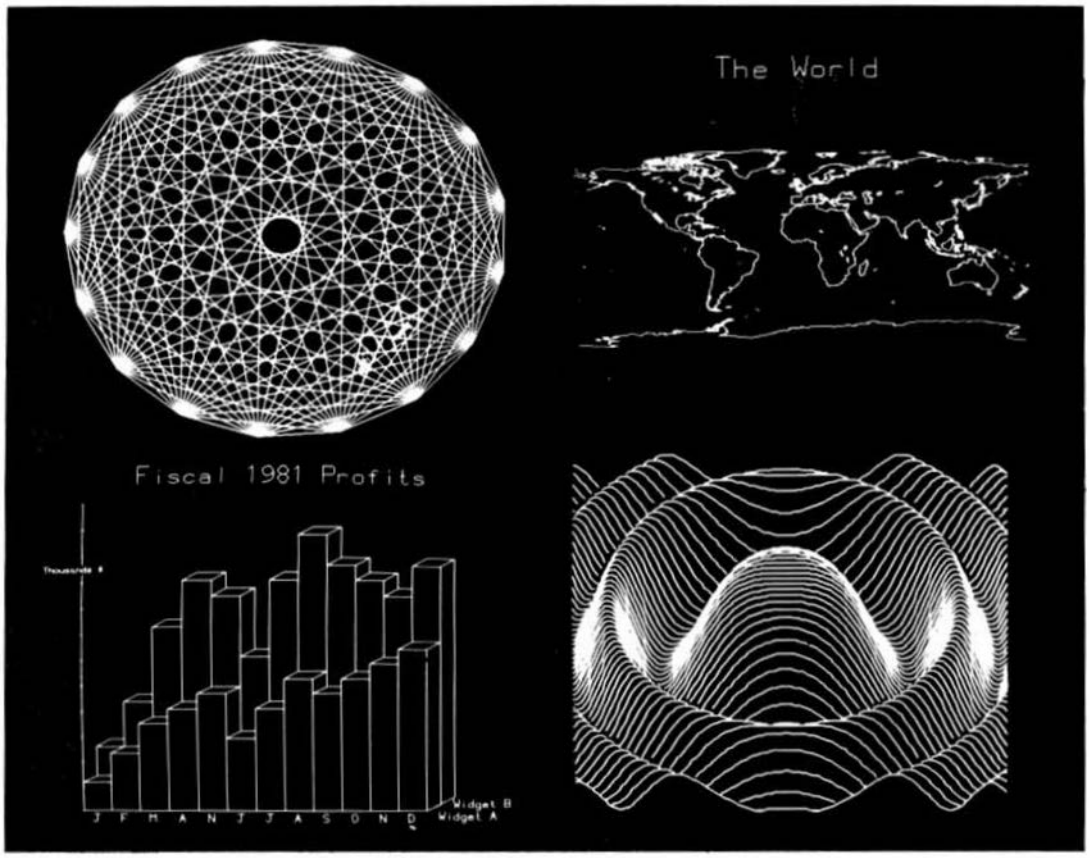
The GIMIX CPU III Card



See Page 64 for Systems featuring this CPU and GIMIX Intelligent I/O Cards

ANNOUNCING ElectroScreen™
the Superior Alternative
to the Traditional Alphanumeric Terminals

only $595



The ElectroScreen™ Intelligent Graphics Board Features:

Graphics
- 512 x 480 resolution bit-mapped display
- Interleaved memory access — fast, snow-free updates

Intelligence
- 6809 on-board mpu
- 6K on-board firmware
- STD syntax high level graphics command set
- Removes host graphics software burden
- Flexible text and graphics integration
- Multiple character sizes
- User programs can be run on-board

Terminal
- Terminal emulation on power-up
- 83 characters by 48 lines display
- Easy switching among user-defined character sets
- Fast hardware scrolling

Additional Features
- SS-50C and SS-64 compatible board
- Board communicates with host through parallel latches
- Composite and TTL level video output
- 8 channel 8 bit A/D converter
- Board occupies 4 address bytes

See your dealer today!

The ElectroScreen manual is available for $10. credited toward purchase of the board.

The ElectroScreen has a 90 day warranty from purchase date.

Dealers, please contact us for our special introductory package.

Privac Inc (703) 671-3900
3711 S. George Mason Dr., Falls Church, Va. 22041

TMP (Total Management Planning) OS-9 SOFTWARE

THESE OFF-THE-SHELF APPLICATION PACKAGES HAVE CHANGED THE PICTURE FOR PROFESSIONAL SOFTWARE USERS

Here are the comprehensively supported OS-9 application software packages that discerning professionals are selecting today. Each one runs stand-alone or in combination with other TMP packages.

TMP/CALC is a *new-generation* "Electronic Spreadsheet" package that goes a quantum leap beyond ordinary Calc software. Superior speed, and extra features like Dynamic Calculations and Dynamic Overlay, mean you can change a cell *anywhere* on the spreadsheet and get automatic updates wherever they belong. No more "wrap-around" since output to the printer is formatted. "Help" screens prompt without erasing data, and the rows and columns on any worksheet are limited *only* by available memory.
ONLY $250

TMP/Manager is a structured database manager, and is the ultimate driving force behind any total *Management/Marketing Planning* system. Its built-in select/sort/merge module — and optimized machine code — makes it superfast, and provides unusually nimble data manipulation characteristics. Other outstanding features are detailed prompting, easy identification of data fields, and an excellent security system.
ONLY $300

TMP FreeForm fills the void between a structured DBMS and word-processing; it's an "electronic index card" package with an endless list of valuable applications such as parts cataloging, look-up inventories and listings of *any* kind. Type anything on up to 32,000 electronic "cards" of up to 9 pages each that reside in "file drawers" (databases). Up to 13 key words can call each page.
ONLY $150

TMP/Front-End integrates any or all of the packages described above, and integrates TMP with word-processors, BASIC and other high-level languages.
NO CHARGE with TMP/Manager

TMP™

TMP PACKAGES ARE EXCEPTIONALLY FRIENDLY. EACH COMES WITH MULTI-MEDIA TRAINING AND OUTSTANDING DOCUMENTATION

AUDIO CASSETTE TRAINING TAPES are provided with each TMP software package. These crisp, professionally produced training aids take a new user through "hands-on" exercises that instantly build confidence, and minimize reference to manuals.

CLEAR DOCUMENTATION is specially prepared for each TMP package. Remarkably readable because they are conversational in style and logically organized, this series of manuals *further* assures quick user comfort and productivity with TMP software. Includes sections on how to best utilize the system; integrating user-written software with TMP packages, and other bonus topics.

VIDEOTAPE PROGRAMS are available to TMP Dealers. They dramatically take the viewer into real business environments for a close look at how a TMP software package is utilized. Video programs can be personalized for specific organizations. Programs may be specified for most popular tape formats.

THE BOTTOM-LINE IS THIS: Software is only as effective as its training and documentation. TMP packages are *incomparable* in both areas.

RUN TMP IN A SMOKE SIGNAL CHIEFTAIN™ COMPUTER TO EXPERIENCE THE KIND OF SPEED AND POWER YOU CAN EASILY AFFORD *TODAY*

Endurance-Certified Chieftains are consistently among the fastest computers in CPU and I/O speeds according to the widely respected BENCHMARK REPORTS. That, combined with almost legendary reliability, makes Chieftains the logical choice in computers using OS-9 operating systems.

Configurations range from floppy-disk systems to multi-user, multi-tasking Winchester hard disk systems with tape back-up; performance approaching mainframes at prices you can live with.

TMP software packages for most OS-9 systems and Chieftain computers are offered exclusively from Smoke Signal. Inquiries from non-Smoke Signal dealers are invited. Call (213) 889-9340.

SMOKE SIGNAL
Chieftain Computers
31336 Via Colinas • Westlake Village, CA 91362
(213) 889-9340

ELEKTRA Means Quality

ELEKTRA CABINET Made of heavyweight 0.090" thick aluminum. Interior is 18-1/2" wide by 21-7/8" deep by 6-3/4" high. Heavy duty A.C. line cord. A.C. fuse holder. EMI filter. Fan with filter. Back panel has 10 cutouts for 'D' type data connectors. Front panel has key on/off power switch, 2 illuminated push button switches (Reset and NMI/Abort), and two cutouts for 5-1/4" disk drives. $250.00

Filler Plate for 5-1/4" drive opening $10.00 Fan Filter $10.00

POWER SUPPLY Highest quality linear power supply CONSERVATIVELY rated at 15a @ 8v, 3a @ 16v, 3a @ -16v. 3 primary inputs for light, rated, and heavy loading.
220v Version: $200.00 110v Version $175.00

DISK REGULATOR BOARD WITH CABLES Standard version for 2 floppy drives. $50.00
Heavy duty version for 1 Winchester drive and 1 floppy drive. $75.00

ELEKTRA UNIVERSAL MOTHERBOARD Heavyweight 0.125" thick, 18" long by 8" wide, 11 memory (50 pin) slots. 4 or 8 slots may be cut off for shortening to 14" or 10" lengths respectively. 8 I/O (30 pin) slots. Complete active decoding of I/O slots. Choice of 4, 8, or 16 addresses per I/O slot. Base address for I/O slots can be placed at 32, 64, or 128 byte increments respectively. 1" spacing between all memory and I/O slots. Extended addressing capability for both memory and I/O ports for meeting SS-50C bus specifications. On board baud rate generator with low and high ranges providing jumper selectable rates of 75 through 38,400 for each of the five baud rate lines. Slow device circuitry permitting 1MHz 30 pin disk controllers to run with 2 MHz 50 pin CPU boards.
Kit w/gold connectors $320.00 Bareboard w/documentation $80.00
Kit w/tin connectors $240.00 Assembled w/gold connectors $360.00
Assembled w/tin connectors $300.00

ELEKTRA CHASSIS Includes cabinet, 110v power supply, power supply cables, standard disk regulator board with power cables, motherboard with gold square pin connectors, assembled and tested. $850.00

ELECTRA CPU 6/9 Use either the 6802 or 6809 (to run 6800 software) or 6809. Has provision for up to 3 27/16 Eproms, 1K scratchpad, MC6840 triple timer, and an optional baud rate generator providing baud rates from 110 through 38,400 baud in two user selectable ranges. The board supports DMA by either HALT or BUSREQ when a 6809 CPU is used. The board will run any of the MIKBUG™ compatible monitors in the 6802/6808 mode and SBUG-E, HUMBUG, and MICROBUG in the 6809 mode. Versions of OS-9™ Level 1 are available.
Bareboard $50.00 Kit $225.00 Assembled $275.00
2K 6800 HUMBUG $40.00 2K 6809 MICROBUG $30.00 see 4K HUMBUGS next page.
Optional Baud Rate Generator $25.00

ELEKTRA OPB DUAL PORT SERIAL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 addresses per port with the B port 2 addresses higher than the A port or for 16 addresses per port with the B port 4 addresses higher than the A port. Each port is terminated at two 16 pin dip sockets, one socket configured for modem and the other for terminal or printer. RTS, CTS, DTR, DCD and DTR are appropriately implemented. Each port has independent selection of baud rate. Each port allows the interrupt request to be independently jumpered to the IRQ or FIRQ/NMI bus line.
Bareboard $20.00 Kit $60.00 Assembled $80.00
Cables (two needed per board) Each $20.00

ELEKTRA OPP DUAL PORT PARALLEL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be used in either the 4 or 16 addresses per I/O slot configuration occupying the first four addresses of the I/O slot. The direction of the TTL buffers can be controlled by either on board jumper connectors or by a signal from the peripherals. The interrupt request lines for each port may be individually jumpered to either the IRQ or FIRQ/NMI bus line.
Bareboard $20.00 Kit $80.00 Assembled $80.00
Cables (two needed per board) Each $20.00

ELEKTRA D-5 Dual drive cabinet for 5-1/4" drives with power supply, line cord, fuse, power switch and power cable to drives 125.00
ELEKTRA HD-5 (Heavy duty version of D-5 package above) 150.00
ELEKTRA SHD-5 (Super heavy duty. Powers 1 Winchester and 1 floppy) 175.00
5" ribbon cable for dual 5 1/4" disk drives 40.00
ELEKTRA D-8 Dual drive cabinet, power supply, pc cable for 8" drives 50.00
8" ribbon cable for dual 8" disk drives 45.00

GOLD 10 PIN CONNECTORS (Specify male w/square pins or female) 1.50
TIN 10 PIN CONNECTORS (Specify male w/square pins or female) 0.50

Prices and inventory are subject to change without advance notice.
This ad is our catalog.

WARNING AAA Chicago Computer Center does not provide repair or diagnostic service for customer assembled kits. AAA Chicago Computer Center does warranty and maintain service for our assembled boards. The customer should carefully take into consideration the small differential separating our kit and assembled prices when making his choice of purchase.

We have introduced our line of computer equipment with the purpose of offering the highest quality of components possible at affordable prices. These products are intended for OEM applications where it is the responsibility of the purchaser to integrate these components with suitable memory, disk controllers, drives, and software along with I/O terminals to form working computer systems.

AAA CHICAGO COMPUTER CENTER
120 CHESTNUT LANE ◆ WHEELING, IL 60090
(312) 459-0450
Technical consultation available 4 PM to 6 PM most weekdays. Closed evenings and weekends.

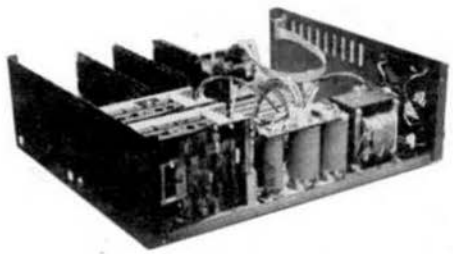
TERMS Minimum order $20.00 Shipping and handling estimates within the Continental U.S., add 3% (MINIMUM $2.50) Illinois residents add 6% sales tax. We will refund your overestimated shipping and handling charges. Foreign shipping and handling, add 10% (MINIMUM $10.00). Foreign orders must be prepaid in U.S. dollars. Heavy foreign items will be shipped air freight collect. Please phone between 4 PM and 6 PM weekdays if questions arise regarding shipping fees. Master Charge, Visa, and American Express honored.

Our apology. We are not staffed to answer technical inquiries through the mail. Please phone for technical help during the hours indicated above. The too frequent changing of our inventory and prices makes it uneconomical to publish a catalog. Our ads are intended to serve that purpose. Prices and inventory are subject to change without advance notice.

SD/DD DISK DRIVES 30 day guarantee	1 head CDC	2 heads CDC	1 head MPI	flippy Siemens	2 heads MPI	2 heads QUME
5-1/4", 40 tracks	225.00	299.00	250.00	260.00	325.00	350.00
5-1/4", 80 tracks	N/A	375.00	325.00	335.00	425.00	450.00
Service Manual (Specify 40 or 80 track)					20.00	35.00
8", 77 tracks DS/DD Qume						525.00
Service Manual (Qume DT-8)						40.00

ELEKTRA

COMPUTER PRODUCTS



SMOOTH™ Software

(All of our software is copyrighted and all rights are reserved. Source is either supplied or optionally available at extra cost so that the purchaser can modify our programs for his own use. Licensing, however, is required for commercial resale.)

SUPER MODEM PROGRAM

Single character commands for simplicity.

Hardware independent (No interrupts required). Assumes modem is connected to an MC6850 (serial interface) and the control terminal is connected to an MC6850 (or MC6820 when used with the video version of GMXBUG™).

Transmit manually to distant computer

Transmit disk files (text) of any length to distant computer

Receive and save disk files (text) of any length on local disk system. If sending computer does not support an X-on/X-off protocol then the received files are limited in size by the computer memory.

Tested for full duplex operation at speeds up to 9600 baud. (CRT terminal must be capable of operating at a baud rate higher than the one the modem is operated at.)

Half duplex option in case distant computer doesn't echo

Echo option so user can simulate a time sharing system. (Super Modem Program doesn't support auto-answer but the source is provided for those individuals who wish to adapt our program to their special needs.)

Replaces CR with CR/LF (user option) for those using time sharing systems that don't transmit LF's

Slow disk file transmit based on character verification (Plus user installed timing loops if necessary) for use on time sharing systems to which disk files cannot be sent at speeds suggested by the baud rate.

Please specify 6800 or 6809, SSB or FLEX™, 5" or 8"
Manual and disk with both source and object code $75.00

STANDARD MODEM PROGRAM

Same as Super Modem Program above but without ECHO option, CR/LF for CR option, slow disk file transmit option, nor X-on/X-off option. Reception of disk files is limited to those small enough to completely fit within the receiving buffer.

Please specify 6800 or 6809, SSB or FLEX™, 5" or 8"
Manual with instructions, source listing, and flow chart; disk with both source and object code $45.00
Manual with instructions, source listing, and flow chart 25.00

MODEMS (By U.S. Robotics) Auto-Dial is Hayes compatible	MANUAL ANSWER	AUTO ANSWER	AUTO-ANS AUTO-DIAL
300/1200 Baud, direct connect	N/A	449.00	499.00
1200 Baud (120 cps), direct connect	349.00	399.00	N/A
300 Baud (30 cps), direct connect	149.00	179.00	N/A
300 Baud (30 cps), acoustic	119.00	N/A	N/A

THESE ADS ARE OUR CATALOG

GIMIX™ CLEARANCE SALE — ALL GIMIX™ ITEMS MUST GO!

All new GIMIX items 20% off list prices while supply lasts. (See our ad in the April issue of the MICRO JOURNAL for list pricing). Some items may be drop shipped by other former GIMIX dealers.

The following used items are in perfect working condition.
32K GIMIX memory boards $175.00, GIMIX 6800 CPU boards $125.00, GIMIX chassis' from $450.00 to $500.00, GIMIX 64 X 16 video boards $100.00, GIMIX 80 X 24 video boards with character generators $250.00.

SMOOTH™ Software

ALL IN ONE
Editor - Text Processor - Mailing Labels
Mailing Lists - Use any CRT terminal and printer

Supports Editing commands such as bottom, change, delete, find, insert (single line), input (multiple lines), list, read, overlay (with cursor editing, character deletion and insertion), overstrike (for selected darker text), print, restart, set, top, underline, up, and verify.

Supports Text Processing commands such as block copy, block move, centering, margin justification (widen and narrow), paging, and tabbing.

Mailing Lists and Labels. Use the same mailing list disk file (with protected areas) for both mailing labels and repeat letters. Repeat letters are personally adressed to each person or selected persons on the mailing list.

Most Powerful File Handler found in any editor. Append one file to the end of another, or insert (merge) one file into another as designated by the line pointer. Print specified lines to your printer or to a disk file. Edit files larger than the text buffer. Does not produce output files when not desired. Delete disk files from the editor.

Printer commands. Control characters can be sent to the printer for format control either directly from the control terminal or by imbedding them in the text. The set command contain interface initialization and character output routines to support the SWTPC MP-C interface as well as the standard serial and parallel interfaces. Jumps are also provided to user supplied printer routines. User selects the port address (0 thru 7, A or B) thereby eliminating the need for the user to install printer software routines. Editor can be initialized for either 4 or 16 addresses per port.

Editor allows exiting to either the monitor or DOS and then reentering (Warm start) without destroying previously prepared text in the buffer. The Restart command erases contents in the buffer without the user having to reload the Editor.

The Editor allows the user to toggle between full duplex (no echo) and half duplex (echo) as needed. It responds to commands in both upper and lower case and can be used to create assembler source code and Basic programs as well as text.

Specify 6800 or 6809, SSB or FLEX™, 5" or 8"	75.00
Printed source listing is available for an additional	35.00
All-in-One, Write'n Spell, and Spell'n Fix package	250.00

Software by Technical Systems Consultants, Inc.	UniFLEX™ w/1 yr. mnt.	FLEX™
DOS (includes Editor and Assembler)	550.00	150.00
Editor or Assembler		50.00
6800 Cross Assembler on 6809	300.00	250.00
6809 Cross Assembler on 6800 or 6800 FLEX™ Utilities		100.00
Text Processor or Sort/Merge Package or 6809 FLEX™ Utilities	150.00	75.00
Extended Basic	200.00	100.00
Basic Precompiler (specify standard or extended)	150.00	50.00
Pascal	300.00	200.00
Debug Package or Diagnostic Package	75.00	75.00
6809 Relocating Assembler & Linking Loader	175.00	150.00
Fortran (Requires Relocating Assembler & Linking Loader)	350.00	275.00
Fortran (With Relocating Assembler & Linking Loader)	450.00	375.00
Cobol	750.00	500.00

Software by Microware Systems Corp.	Run-Time Package	Update	Source	Manual Only	Object w/Man.
OS-9™ Level One Operating System		75.00	400.00	40.00	200.00
OS-9™ Level Two Operating System		75.00	N/A	40.00	500.00
BASIC 09™	100.00	75.00	N/A	25.00	200.00
OS-9™ Macro Text Editor			300.00	15.00	125.00
OS-9™ Interactive Assembler			300.00	10.00	125.00
OS-9™ Interactive Debugger (Disk version)			100.00	10.00	50.00
CIS Cobol Compiler	400.00	50.00	N/A	80.00	900.00
Pascal Compiler	100.00	100.00	N/A	40.00	400.00
Microware yearly support service ($200.00 for OS-9 Level 2)					75.00

SPECIALS

- New Televideo Z80A, 4MHz, 64K computer system with 950 compatible built-in terminal, dual 5-1/4" floppy drives (1 Mbyte capacity total), CP/M — 3295.00
- Same but with one floppy and one 10Mbyte Winchester — 5495.00
- Six user versions available — Call
- Modems (Up to $100.00 off list). See our ad on the previous page.
- SWTPC MP-09B, socketed, with baud rate generator (only 1 left) — 175.00
- SWTPC MP-A2, socketed, all with crystal clock 1 MHz (only 4 left) — 115.00
- SSB BFD Floppy DiskControllers (Version 3) — 175.00
- MICROBUG (2K, 6809, Baby HUMBUG by Peter Stark) — 30.00
- Cat 300 Baud Accoustic Modem (only 1 left) — 99.00
- SWTPC 4K Memory $15.00, 8K Memory $40.00, MP-C $10.00
- SWTPC Chassis,PS, MB combos (some with modified PS) — $50.00 thru 125.00
- High speed tape reader — $75.00
- SWTPC PR-40 Printer — 75.00
- Percom 50 pin prototyping board — $30.00
- 30 pin proto. board — $20.00
- Used MPI B51 and Siemens 5-1/4" floppy drives — 125.00

Special Software

MICROBUG (2K, 6809 Baby HUMBUG by Peter Stark	30.00
4K 6809 HUMBUG	75.00
4K 6800 HUMBUG (RAM needed at $A000 and $D000)	8.00
2K 6800 HUMBUG (With cassette LOAD and PUNCH)	40.00
2K 6800 HUMBUG (Extra commands instead of cassette software)	40.00
Other HUMBUG versions including video versions are available.	
Spell'n Fix by Peter Stark	178.58
Write'n Spell by Peter Stark	75.11
All-In-One, Spell'n Fix, and Write'n Spell package	250.00
Dynamite Disassembler	60.00
Dynamite + (Update. Send original diskette plus $40.00)	100.00
SUPER TEUTH Disassembler System ($101.00 for O -9 version)	99.00

SWTPC

6809 SWTPC FLEX™ Disk and manual (Disk only 15.00)			35.00
DC-4 Disk Controller (SS/SD, SD/DD, 5-1/4")			230.00
532 Universal Static Memory Board (without memory chips)			124.50
SBUG-E (2716 compatible)	25.00	MP-R 2716 Eprom Programmer	114.50
MP-S2 Dual Port Serial	120.00	MP-L2 Dual Port Parallel	120.00
MP-N Calculator Board (kit)	54.95	MP-N (assembled)	92.00
MP-T Interruot Timer	92.00	MP-09 2MHz 6809 CPU Board	295.00

Smoke Signal Broadcasting

DCB-4A Double Density Controller Board for 5" and 8" with DOS					549.00
SSB DOS (Specify 6800 or 6809, BFD or DCB-4A, 5" or 8")					75.00
SE92/SA92-5 (6809 Edit/Assm for DOS)					89.95
SSB Monitor (Specify 6800 6809, $8000/$E000/$F7E0)					75.00
SSB version of FLEX™ (without Editor and Assembler)					150.00
LMB-1A Motherboard					399.00
SCB-69 6809 CPU Board					399.00
PAR-1 Dual Port Parallel Board					89.00
ER-2 Dual Port Serial Board with 2 Cables					129.00
Chief 90 64K Computer System					2195.00
Chief 9524 64K Computer System with DS-DT-DD 5" FD					4325.00
Static Memory Boards	M-16-X 195.00	M-24-X 295.00	M-32-X 395.00		
Dynamic Memory Boards	M-128-X 995.00	M-256-X 1295.00	M-512-X 1895.00		

THE BEST 30 PIN FLOPPY DISK CONTROLLER THAT YOU CAN BUY! — THE ELEKTRA SFC

Controls up to four 5 1/4" drives and four 8" drives for a total of eight system drives for both 6800 and 6809 systems. (FLEX software has a built in limit of a system mix of four drives). Improved programmed input/output (PIO) interface that allows single and double density 5 inch operation at 1MHz in both 6800 and 6809 systems. Allows single density 8 inch operation at 1MHz in both 6800 and 6809 systems. Allows both single and double density operation of both 5 inch and 8 inch drives at 2MHz in both 6800 and 6809 systems without slow device circuitry. Hardware and software compatible with existing SWTPC DC-1, DC-2, AND DC-3 type controllers. Controls virtually any floppy disk drive, any mix of drive size, recording density, track density, number of sides, motor control, with or without head load solenoids, including the new 2 Mbyte 5 inch drives. Analog phase locked loop data separators for maximum reliability. Provides separate adjustments for 5 inch and 8 inch drives. Analog write precompensation circuit with separate adjustments for 5 inch and 8 inch drives. Designed to meet the data hold requirements of Western Digital floppy disk controller IC. Schmidt trigger hysteresis input buffers for disk drive signals. Software write protect control. Interrupt/Data Request enable control. 12 option jumper areas to select just the right combination of hardware features to match your drives and software. Provides many separate option selections for 5 inch and 8 inch drive operation. Gold bus and drive cable connectors. Test connector for easy setup and field service. Single 5 volt power supply. 4 or 16 addresses per slot decoding. Ideal replacement for existing controllers to add double density data storage and 8 inch drive capability. Disk driver software and disk formatting utilities available to extend the capabilities of your current software. Get more for less with the ELEKTRA Super Floppy Controller.

ELEKTRA SFC Super Floppy Controller (assembled and tested)	$ 50.00
Disk with drivers and formatting utilities	$30.00
EPROM with driver overlays	$30.00
TSC General FLEX or FLEX 9 both with Editor and Assembler	$150.00

THE BEST WINCHESTER DISK SYSTEM THAT YOU CAN BUY! — THE ELEKTRA WINCHESTER

Has automatic error detection and **CORRECTION** of up to 11 bit burst errors.

SS-50 bus, extended addressing capabilities, DMA, on board sector buffer, operates with 6800 and 6809 type processors, drivers included for FLEX 9 or OS-9. Drivers for FLEX 2 (6800) available for an additional $100.00. Price includes host interface, controller, drive(s), and cables.

7.5 Megabyte ELEKTRA single drive Winchester system	$1995.00
15 Megabyte ELEKTRA dual drive Winchester system	$2995.00
19 Megabyte ELEKTRA single drive Winchester system	$2995.00
38 Megabyte ELEKTRA dual drive Winchester system	$4995.00
ELEKTRA heavy duty regulator bd. (powers 1 floppy and 1 Winchester)	$75.00
ELEKTRA SHD-5 cabinet, ps, ps cables (powers 1 floppy and 1 Winchester)	$175.00

AAA Chicago Computer Center. See our ad on the previous page to your left for ordering instructions.

Technical Consultation available 4 p.m. to 6 p.m. most weekdays.
Closed evenings and weekends.

Flex User Notes

Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

Heating Control

Last time, we had started the discussion of a home heating control to illustrate the use of a 6809 system to perform a control function. We had gotten as far as a program to use the JPC A/D board to read temperatures and temperature commands on up to 8 rooms (16 channels) and display the readings and commands on the terminal. We also showed the schematic for the hardware required for that portion of the control.

In order to keep things fairly simple, I am going to opt for a bit more hardware on the output side than is absolutely necessary, in order to keep the software simpler. If you follow what is presented, you can undoubtedly adapt the idea to whatever sort of actuator and sensors you have available.

Safety (or failsafe operation) must be considered as a primary feature. The original thermostat will be left connected. You may set it to a temperature lower than you will want in the house while you are present. If the house power fails, of course the gas valve can't be turned on while the power is off. However, the return of power must not cause the system to call for heat. Unfortunately, the 6821 PIA can supply no source current when is is used as an output, so a pull up resistor is required. When it is reset, it comes up as an input, and the pull up will cause the output to go high, making it look as though it were on. We must therefore provide a "latching relay" that will drop out on power failure and disable the system until it is reset manually after power returns.

Such a feature won't be necessary if you plan on using a single board computer and have the program in ROM, because power restart will start the program and re initialize the PIA used in the output, as well as the one on the A/D board. However, if you use a system in which the program is loaded to RAM from a disk or tape, the manual reset is absolutely necessary. The thermostat will take over and prevent the pipes from freezing, etc.

Probably the best bet when the system is run from RAM is to resort to the manual thermostat when you are not home, anyway. I've spent time and words on this problem because it is typical of a machine control application. When the power fails or is shut off by an "Emergency Stop", the control must return to a "benign" state so that the reapplication of power leaves it in its "Manual" mode, and with no spindles or part transfer mechanisms in motion. An error on the part of the programmer in that regard could cost a maintenance technician a finger, hand, arm, or worse, his life.

Now, on to our scheme. We will use one port of a PIA to provide 8 outputs. These outputs will signal "Baffle Open" or "Water Valve Open" when high, and the reverse when low. Such an arrangement is quite simple to interface with a solenoid valve, since all that is needed is a power amplifier or a relay for each valve. In the case of a motorized baffle in a heat duct, the control means for reversing the motor, and sensing its travel limits with two limit switches, must be supplied. See the diagram for the control arrangement. Most heat control is done with 24 volts AC which is not dangerous to work with. Some units use 120 volts, however, which is VERY hazardous. In any case it is good practice to turn off the control power when working on the system.

The software must not call on the furnace to turn on and off at too frequent intervals. We must therefore provide for several possibilities. It seems to me that a good way to minimize cycling is to provide for a couple of degrees "hysteresis" in the control. That is, if the temperature is set for 70 degrees in a room, the control will supply heat until the temperature reaches 71, and will not again call for heat until the temperature reaches 69. Depending on your installation, you might want to widen those limits a bit more. It might also be prudent to require at least two zones calling for heat before turning on the furnace. A system set to supply air sufficient for ducts to 8 rooms, might overheat if all but one duct were to be shut. An alternative to this is to set the "closed" limit on the baffles such that all are open to about 20% of their wide open air flow.

Furnace controls have overtemperature sensors that turn the flame off but leave the air on. If you install this system, you should check the overtemperature limit to see that it is not being reached when only two ducts are open. If that is a problem, you might want to require three zones calling for heat before turning on the furnace. Control will still be vastly better than that obtained with a single thermostat in the living room.

I've added the necessary software to put the readings in an array and then perform the necessary logic to open or close the various zone ducts or valves. The additions include a section in the initialize routine to set up the PIA properly, and the logic to open ducts or valves and call for heat. The heat contact should be wired in parallel with the normally open contacts of your house heating thermostat. Note that I've used both ports of the PIA as outputs. MP-LA cards will require jumpers to select output buffering on both sides of the port, and the old MP-L cards will require foil cuts and jumpers to do the same thing. The newest systems have programmable buffer data direction registers included in them. See your parallel port instruction manual for the necessary programming instructions, and include them in the initialize routine. I realize that it would be possible to use one side of the port, using 8 bits for control and a handshake line for the heat switch. I've done it the way I have for simplicity.

The final program has been done in the Pascel version only. The first change was to reassign the channel numbers for the A/D converter in an order that simplifies the handling of them later in the program. I've added an array TEMPS to hold the measured temperatures and set points for each room. The new procedure CTRL_TEMP first reads all the channels and puts the readings into the TEMPS array.

I also created an array ROOM_STATUS that contains the heat ON or OFF information for each room. This array is necessary because we only want to turn the heat on if the room temperature is a degree below the set point, and off if it is a degree above. That feature is the hysteresis mentioned above. The second loop in the CTRL_TEMP procedure compares the temperatures and set points for each room, and sets the ROOM_STATUS accordingly.

The last loop might bear some explanation. We need to "assemble" the room status information into one byte to write to the output port to turn the proper valve or baffle controls on and off. The loop runs from "7 DOWNTO 0" so that we can get the value for room 7 into bit 7 of the control byte which will be assembled in the variable VALVE. The loop counts the number of rooms that require heat, in the variable COUNT. If a room requires heat, a 1 is added to VALVE. If not, nothing is done to VALVE. In each pass through the loop, after adding one or not, the value is multiplied by 2, which is the same as a left shift by one place. After finishing the loop, the value of VALVE has a 1 in each place where a valve should be on, and a 0 for off. If you don't quite see that, a bit of diddling with a pencil and paper should clarify the point.

This is a useful technique for many sorts of control applications. The inverse may be done, by operating on a value input from a port. VALUE MOD 2 will yield a 1 if the low order bit was a 1, and 0 if it was a zero. Then VALUE DIV 2 will put the next bit in the low order position so that MOD will get the next bit, etc.

The main program remains essentially unchanged. I simply added the call to CTRL_TEMP at the proper place, and changed the write statements so that they get the various temperatures from the TEMPS array rather than reading the A/D again.

Note that I've tried to keep the program understandable by not using some of the more advanced features of Pascal. I've assigned room names constant values so they could be used as array subscripts. This technique could be used in a BASIC program also though admittedly, two letter room names might be harder to understand. This program will fit a couple of 2716's in a stand alone system.

IMPORTANT NOTE: Neither I nor '68' Micro Journal will be responsible for any damages that might result from the use of this program or one patterned after it. The user will take full responsibility for the fitness of the program for the application to his heating system. If you attempt such a system and are not completely familiar with furnace controls, consult a heating contractor to be absolutely certain that you do not defeat the safety controls on your furnace. The reason for presenting this particular program was to present a reasonably simple control application with which most readers might easily relate. I personally have not built such a system, and presently don't intend to do so. I don't guarantee that there are no bugs in the program as presented here.

More on PL9

I have continued correspondence with Windrush Micro Systems regarding their PL9 compiler. I've found a few more bugs, though fairly obscure ones. Windrush has responded by return mail (as quickly as that is possible round trip to England), and in general the fixes have been a matter of a few bytes of patch. All patches have been completely successful.

I received a letter yesterday from Graham Trott, the author of PL9. I feel that we are "kindred spirits", having both gotten into computing via the simple single board microprocessor system. I have an observation (I'm editorializing here) that those who enter computing from that direction tend to think simple. Graham has written a compiler about as capable as most of the Pascal and C implementations, though admittedly requiring a bit more thought on the part of the programmer. His compiler is a single file containing the compiler proper, and an editor roughly as capable as the TSC EDIT program. It is only about 60 sectors (less than 15K)!

I have found that those who have entered computing via experience with mainframe computers, generally have trouble thinking small enough to get a compiler into "only 56K". Roughly quoting one supplier of a "tiny" C compiler, "The compiler occupies nearly all of 56K of memory, and we therefore have no plans for implementing further features of C in the future". This company's "printf" library routine was so inefficient that it generated more code than the entire runtime package.

Funny how some simple minded folks can do the impossible (like the chess game that runs in a KIM-1 single board computer in 1K of RAM). To misquote something I saw in the third book of Murphy's Laws: To write a very complex program is an easy task. To write one that is simple is a very complex task.

Off my soapbox and back to facts... I have written a couple of significant programs in PL9. I found that the first, which runs about 8 pages of source, compiled in 32 seconds, including output to a binary file. It generated all of 5.6K of output code, or about 700 bytes per page. A good Pascal version generated about 1K per page. I have to report that this is the most efficient compiler I have found to date. That is partially true because of the easy access the programmer has to the program. He may do a number of things to optimize the code generation that are not possible with many of the more standard languages. Perhaps some day I will print some more specific examples here.

Gimix DMA

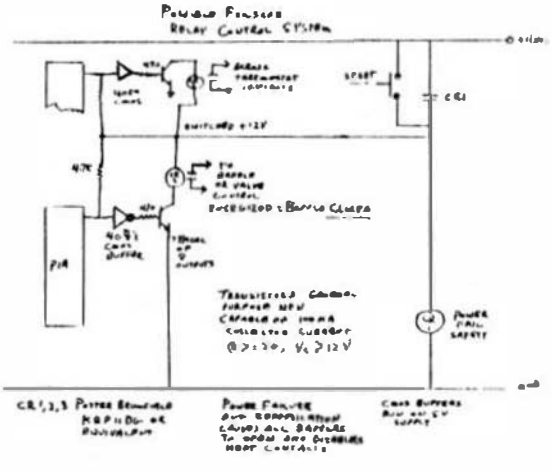
I have made an arrangement with a friend to try out a GIMIX DMA disk controller board. We traded temporarily, my SWTP DMAF2 for his GIMIX. I had one major trouble. All my disks were formatted via SWTP's "Extra Density" format, which yields 4408 sectors on an 8" disk, by squeezing two extra sectors on each track. Gimix FLEX wouldn't read those disks, and I had to borrow my DMAF back and format all my disks in the normal double density mode. I of course took advantage of that to get my disks organized a bit. Since formatting and copying take so long, I was able to get my computer area cleaned up and work on the basement a bit for most of the day during which I did the copy.

At any rate, once I had compatible disk formats, I found the GIMIX board very nice to use. I was able to connect two 5" drives along with my 8" drives, and for the first time, have four drives present on boot up and all capable of double density. With this board and GIMIX FLEX, the user can format both five and eight inch disks in any format, ie. single density, double density, single or double sided, and any number of tracks up to the limit for a particular disk size, ie 80 tracks for 5" disks and 77 for 8". That brings up an interesting question. Why hasn't someone come up with a double track density 8" disk drive, which would hold 2 Megabytes in the DSDD mode? Perhaps 8" drives will soon be obsolete. The 80 track DSDD 5" drives already are capable of about 750K bytes of storage. Anyway, I have been using the GIMIX board for a month or so now, and I don't recall a single retry in reading or writing a disk file.

Forth Again

I just received my April '68' in which the letter from Gary Bergstrom and my reply were published. I received another letter from Gary a few days ago. I must say that he is persistent and factual in his replies. All I am going to report here is that Gary has clarified what he means by extensibility to me, and that neither of us has changed our opinions by very much. By now you have all got ten tired of the discussion, and I will say little more on the subject. We'll leave it to the people who are really enthusiastic about Forth to write some articles about it.

- - -



OS9 USER NOTES

By: Peter Dibble
517 Goler House
Rochester, NY 14620

A large number of the exciting things that can be done with OS-9 involve processes. Every program running under OS-9 is a process. Each process runs as if it had the machine to itself (except for speed). When a new process is started, OS-9 loads the Program module for the process if it isn't already in core, creates a Process Descriptor for it, allocates the necessary amount of memory, gives it standard input and output files, and lets the new process go. One of the ongoing tasks of the operating system is to divide processor time between all processes so that the system's resources are used as efficiently as possible, and all the processes run without too many noticeable jerks. You can tell OS-9 to favor a process by giving it a high priority (with the SETPR command), or you can give a process a low priority if you don't much care how quickly it runs.

A new process is created with the OS-9 service request F$Fork. Before issuing this service request you must set up the registers as follows:

X Address of the name of the module you want to FORK or the file that contains the module.

Y The size of the parameter area.

U The beginning address of the parameter area.

A The Language/Type code. That is, the type of module you want to fork. Basic09 has to be treated differently from object code.

B The amount of optional storage to give the new process. The interesting thing here is that it is possible to send a block of data to the new process using the parameter area. The new process will be started with X pointing to the start of a copy of the parameter area and D containing the length of the parameter area. In languages other than assembler, the parameter area can be found by noting that the parameter area is the place where the shell places the command line parameters for a program. The shell usually starts programs by FORKing them, so in any language, if you can get to the command line parameters, you can get at parameters passed through Fork in the same way.

By using the parameter area you can pass a lot of information to a new process, but you can't get anything back through the parameter area. Remember that the parameter area gets copied into the new process's address space. It is like a Pascal pass-by-value parameter -- changes don't get back to the invoking process. Still, for many jobs, the one time, one way communication afforded by the parameter area is sufficient.

Neither Basic09 nor Pascal has all the necessary functions for dealing with forked processes, but they can be reached through assembly language subroutines. I have included two short assembly language subroutines which should help. StrtTask, and WaitTask are meant to be called from Basic09, though modified versions could be called from Pascal or any other normal language. StrtTask starts execution of a process, and WaitTask waits until a child of the calling process completes before returning to the caller. These aren't examples of elegant coding, but they are good enough to play around with from Basic09. The Basic09 programs Driver, and BTest are respectively a driver for the assembly language modules and a stub for testing them.

StrtTask is an interface between a Basic09 program and the OS-9 Fork service request. Normally, a fork is done with the SHELL statement in Basic09. By using StrtTask instead of SHELL to start "child" processes, a program can gain better control of the parameters. StrtTask allows full control of the F$fork system service request.

The first parameter which StrtTask expects is the name of the module to be started. It should be passed as a character string with a terminator, such as a space or carriage return, after the last character of the module name. If the module might not be in memory, the name of the file which should be loaded to get the module should be the first parameter instead of just the module's name. The F$Fork system service request description in the OS-9 System Programmer's Manual has more details about this, and all the other parameters for StrtTask.

The second parameter is the process number of the new task. It is a byte field which need not be initialized. StrtTask will place the process number of the newly started process in this byte. This is the only parameter which is returned from StrtTask. The process number is useful if you want to communicate with the new process, or to wait for a particular process to complete.

The third parameter is the language/type byte which describes the module you want to run as a child process. The easiest way to discover the proper value for this byte is by checking the module you want to fork. You can see the language/type byte for a module by loading it and doing a MOIR E command, or by doing a IDENT command on the file the module is in. Remember that this byte is displayed in hex. Object code programs (generated from assembly language) generally have a language/type byte of $11, or decimal 17.

The fourth and fifth parameters are the length of the parameter area to be passed to the forked process, and the parameter area itself. The parameter area can be any type of data you want to pass to the new process. The length of the parameter area is passed as an integer. If you invoke a module which is usually started from the shell, the parameters should be a character string terminated with a carriage return. If you want to invoke a module which runs under Basic09, it is particularly important to include the carriage return at the end of the parameter area (which contains the name of the Basic09 I-code module to run and any parameters for it). Strange things happen if you don't.

The last parameter is the amount of optional storage space you want to give the new process. This is the number usually placed after the "" on a shell command line. The number can range from zero to 255 (it is a byte field), and may only be in units of pages, not Kbytes.

If the fork service request itself gets a bad return code, it will be returned to the calling program as an error. In general the new process will still be running when StrtTask returns to the calling program, so there is no way to know what the completion code of the new process is (going to be).

Sometimes you may want to start a process going and continue without waiting for the new process to complete, but you may need to wait for it to complete at some point. This is where WaitTask comes in. WaitTask will wait (just sit there) until one of its children (a child of the program that called WaitTask) completes. If there are several children, the first one to complete will let WaitTask return to its caller. If there are no children, WaitTask will return with an error. If a child process terminates before it is waited for,

its process descriptor will linger around in memory until a wait is done by the parent process.

WaitTask has two parameters, both of which are set by WaitTask. The first parameter is a byte containing the process number of the process whose completion let WaitTask return. The second parameter is the completion code of that process. If there are several children that might terminate, the process number parameter can be used to cause the calling program to keep calling WaitTask until the necessary process completes.

To use this package of modules (StrtTask, WaitTask, Driver, and BTest):

Assemble a file containing StrtTask and WaitTask
```
asm StrtTask o 24k
```
Save the packed form of BTest
```
BASIC09
**** type in BTest
save
pack
```
load StrtTask and WaitTask
```
load StrtTask
```
or if you are still in Basic09
```
$load StrtTask
```
Type in Driver $the basic driver programt
run Driver

There are a lot of interesting things that can be done with these modules. You can fork any program you want, not just packed Basic09 modules, but the special features of the shell, such as I/O redirection, aren't provided by StrtTask. You don't need to wait for the new process to complete, but if the new process does I/O to standard paths, it can be very hard to tell what is going on on the screen. If you haven't made a mistake that causes several processes to use the terminal for I/O at the same time yet, you should. It is educational.

The thing about new processes that particularly excites me is that under Level Two each new process gets a new address space with up to 64K. The main problem with the modules included with this column is that there is only one-way communication with forked processes. The parameter area goes from the parent to the child, but the child only sends a completion code back to the parent. There are easier ways to communicate. We'll get to them later.

DYNACALC (REVIEW-OS9)

by Peter C. Dibble
517 Goler House
Rochester, NY 14620

Overview

DYNACALC is a very capable electronic spread sheet program. It is enough like all the other spread sheet programs (visicIones) so anyone familiar with one of them should be able to adjust to DYNACALC very quickly. It is a very good exemple of the current state of the art.

A electronic spread sheet program makes the terminal appear to be looking at a section of a large grid. The "cells" in the grid can each contain a number, equation, or character string. The equations usually operate on the contents of other cells (A column of cells might contain monthly expenses, and another cell somewhere on the grid might contain the sum of all the cells in the column of expenses.) The special thing about electronic spread sheets as opposed to paper spread sheets is that when a number or equation on an electronic spread sheet is changed, all the cells that depend on that value are updated to reflect the change. This is a simple idea, but such a good idea that I know of many people who have purchased computers just to be get at this kind of program.

Some Details

DYNACALC is a large 6809 assembly language program which has been available under FLEX since last year. It is now also available under OS-9. The OS-9 version doesn't seem like warmed over FLEX code; it seems to have been designed for OS-9. It is reentrant (the same module can be used by any number of simultaneous users), and uses standard input and output. Practically any CRT type terminal can be supported. In fact, the warranty for DYNACALC says that if you have a CRT terminal with at least 80 characters per line and direct cursor addressing:

> If your terminal has the required characteristics, but you are unable to configure DYNACALC to work properly (using the INSTALL utility), send us your original DYNACALC diskette and a copy of the operator's manual for your CRT. We will either make it work on your terminal, at no extra charge to you, or refund your full DYNACALC purchase price.

That is a very impressive commitment! If you have several users on your system with different types of terminal, you can get DYNACALC to support them all concurrently if you have each terminal type use a different data directory, and put the appropriate terminal file in each directory.

DYNACALC can save the contents of a spread sheet in a file that can be read by other programs. I wouldn't call the files easy to use, but they aren't impossible to use either, and the format is clearly documented. DYNACALC's saved data is hard to use because the format of the file reflects DYNACALC's flexible attitude towards the user - it will take any sort of data scattered around anywhere you like. If you want to create a file for DYNACALC to use as data for a spread sheet, you don't have to cope with the vicissitudes of humans. It is a relatively simple job to create data files for DYNACALC.

An excellent help facility is an integral part of the program, though you can remove it to save space if you want. Most of the time you can type a "?" to access a screen of terse explanations of your options. The help screens do not take the place of reading the manual, but they can provide a quick jog of the memory. There are also 12 error codes which I wish all visiclones had. Spread sheets can take on some of the attributes of complicated programs, especially hard to find bugs. Imagine trying to debug a program with only one error message like "Sorry, I can't do that," "Say What?" or whatever.

My copy of DYNACALC came with terminal files called:

ct_82	ct_82_92	c8200	c8200_92
c82_w	c82_w_92	h_1400	h_1420
h_1500	adds_vpt	adds_3a	h_19
act_iv	adm_3a	ansi	tv_912
pe_550	info_100	iq_120	tv_950

I recognize SWTPC, Hazeltine, Adds, Heathkit/Zenith, ADM, and Televideo in there. Even if your terminal isn't in that list, you can use the INSTALL.DC utility to build a terminal file for your terminal.

A particularly strong point of DYNACALC is the set of powerful functions it supports, including basic math (trig, log/exp, square root, max/min, pi, int, round, and absolute value), "group" functions (sum, average, standard deviation, net present value, choose, lookup, and index), and a bunch of miscellaneous functions. Choose selects the nth entry from a list, lookup is the standard visiclone lookup function, and index is like lookup except that it scans for an exact match instead of greater than. Many of DYNACALC's functions work with either character strings or numbers. This expands the usefulness of the functions substantially.

DYNACALC has commands which move rows and columns around, and do insert and delete operations on them. The fanciest command in this family is the sort command, which allows you to sort rows or columns based on the values in a column or row respectively.

I have never been entirely pleased with the speed of any program. Of course I wish DYNACALC ran faster, but I don't remember using a spread sheet program on a microcomputer that ran faster.

Limitations and Problems

The only real problem with DYNACALC is with its terminal support, and I'm not sure it could have been done much better without losing generality. The terminal support problem is not a major one. In fact, I imagine that after a few months of using the program I will feel nothing but affection for it.

It is hard to choose characters to use as arrow keys. DYNACALC uses curly and square brackets as cursor control keys by default. This is a good choice if you want to drive it with a disk file, but not very intuitive. If you like this choice as little as I did you can change it with INSTALL. Unfortunately Install only allows you to use single characters as control keys; my terminal, like most terminals, sends escape sequences when the arrow keys are pressed.

Screen updating is not as fast and smooth as it is on machines that have integral screen support. I understand that a 9600 baud terminal can't possibly compete with memory mapped video, but I believe that, if the insert and delete character and line facilities on my terminal were used, the screen could be updated more quickly. It would have been hard to make DYNACALC support more advanced terminals while still supporting "dumb" terminals, but I wish it had been done.

Summary

DYNACALC is a fine program, but although it seems to have been written by a programmer familiar with OS-9, it doesn't make the fullest use of the power of OS-9. I wish DYNACALC could use all available memory instead of just 64K, and I wish printing was handled by a separate process so I could start a copy of a sheet printing, then continue work on the original. Extended memory probably could have been used under Level Two without degrading the program under Level One, and multiple processes are supported by both levels of OS-9.

I find myself expecting a great deal of DYNACALC. My carping at its terminal support (which is in many ways unusually good), and pushing for support of fancy OS-9 features is a reflection of my very high opinion of the program.

I know people who find it reasonable to buy a personal computer just to have an electronic spread sheet. DYNACALC is an excellent spread sheet program. It can help with any number of business problems, simple problems in the sciences, and just plain showing off the computer to the uninitiated. I think DYNACALC is a program which should be included in the toolkit of most OS-9 users. One warning, spreadsheet programs tend to be popular. I am afraid that I will have to wait for a crack at my machine more often now that I have DYNACALC on it.

DYNACALC, all versions OS9™, FLEX™ and UniFLEX™ version are available from:

SE MEDIA
PO Box 794
Hixson, TN 37343
(615) 842-4601

- - -

Editor's Note: It should be noted that a lot of the work of creating a spreadsheet can be eliminated by the use of the many spreadsheet 'how-to' books, available at most computer book outlets. DYNACALC is so 'like' the other popular programs that the examples are very easy to carry over. Fact is you will have an advantage, DYNACALC does MORE than most of the others. In my personal opinion, DYNACALC is one of the most powerful programs to ever run on any computer. We know of many Standard S50 Bus computers that have been sold just to use DYNACALC. Should tell you something.

DMW

O-F (REVIEW - OS9 TO FLEX OS9)

by Peter C. Dibble
517 Goler House
Rochester, NY 14620

A few weeks ago I spent most of a Saturday hooking my old SWTPC FLEX machine to my new machine as a remote computer so I could use it to write a FLEX-format disk. It felt rather odd using my "smart terminal" program to communicate with a machine less then a foot away. The process involves shuffling disks drives back and forth, and much opening and shutting of cabinets. I don't like it much. My new machine has GIMIX software switching, so I can run FLEX on it, but even the remarkable GIMIX CPU board can't run both operating systems at once. On occasion I have uploaded a file from one OS to an IBM and then downloaded it with the other OS, accomplishing a change of disk format from FLEX to OS-9 or vice-verse. These methods are all inelegant, ad hoc solutions to a problem. Dr. Matthew Scudiere has come up with a much cleaner solution: He has written an OS-9/FLEX copy program called O-F.

General System Description

This OS9/FLEX copy program is a BASIC09 program which allows the user to convert an OS-9 format disk into a hybrid form which can be read and written by FLEX. In the process of doing this it makes the disk inaccessible to OS-9 except as an entire disk (i.e. /Dn@) but O-F is able to copy files to and from the hybrid disk, and read the FLEX directory. The disk that results from the reformatting is enough like standard FLEX format that FLEX doesn't know the disk isn't one of its own.

Limitations

Only freshly formatted, single sided 5 or 8 inch disks with no bad sectors can be used, and there is no way to use a disk which is in real FLEX format (formatted by the FLEX NEWDISK or FORMAT program). The FLEX to OS-9 copy part of the program expends tab characters into strings of blanks by default, but there is an option which causes the file to be copied intact. Of course, this program doesn't make any attempt to convert FLEX programs into OS-9 programs. That is work for other programs.

Operation

In order to run O-F you must first start Basic09. The version I tested was in source form, so I had to load it and run it. If it is distributed as Basic09 I-Code it should be possible to just run it. The program lists 7 options:

0 Directions
1 FLEX Directory
2 Copy FLEX text file to OS9
3 Copy OS9 path to FLEX
4 Delete FLEX File
5 Reformat OS9 Disk
6 Exit program and prompts for a selection.

"Directions" produces a quick summary of the function of the program, about half a screen full. "FLEX Directory" lists the basic information in the directory of a pseudo-FLEX disk: file name, Begin, End, Size, and date. It also gives the number of sectors used on the disk, and the number of sectors left. The "Copy FLEX text file to OS9" dialogue is:

FLEX Compatible source Drive ID --
FLEX file name to copy --
Copy to OS9 destination path -- The "Compatible source Drive ID" is the device name for the disk that has been reformatted; that wasn't too clear to me. The "Copy OS9 path to FLEX" dialogue is:
Drive ID --
FLEX File name to write (Use upper case) --
Copy FROM OS9 SOURCE path -- To delete a FLEX file, select 4, then:
Flex compatible source Drive ID --
FLEX file name to delete (use proper case) --

The dialogue for reformatting a disk is very cautious:

Drive ID --
Are you sure? --
Overwrite -- <old volume name here>
Are you sure? --
5-in or 8-in disk?

I tried reformatting and writing on 5 inch disks (SS/SD, SS/DD, 40 track and 80 track), and 8 inch disks of all permutations. It worked on the 5 inch disks, and on SS single and double density 8 inch disks. I was able to read pseudo-flex files created by O-F from FLEX without any trouble. O-F had no trouble reading files written by FLEX on disks reformatted by O-F. The reformatted disks were also fully usable in FLEX. FLEX truly thinks the reformatted disk is one of its own. One nice touch is that the program puts two entries in the OS-9 root directory of the reformatted disk:

** NO OS9 Files Allowed ** (This is a FLEX copy disk)

These entries appear if you do a DIR command on the reformatted disk, letting you know very quickly that this disk is special.

Evaluation

This is a competent and very useful program. It is especially well equipped with error messages and informative text. In fact, although it came without a manuel, I was able to follow the built-in directions without any trouble. I do hope that a manual is available by the time this program hits the market. A program without a manual seems somehow unbalanced even if it is usable without documentation. A nice extra is that it appears that this program may be distributed in source form.

O-F works by tricking FLEX. This together with the variety of disk formats that FLEX might use forces the program have some odd restrictions. The most serious limitation is the restriction to specially formatted disks. It certainly would be nice to be able to drag out a four year old FLEX disk and read it with this program. The restriction to single sided disks is reasonable in the context of copying files from one format to the other. For some people the most important limitation will be the language requirement. Since this is a Basic09 program, you must have Basic09 to be able to run it. It could be a measure of the desperate need for a program like this one that it is being hustled out in Basic09 form.

One of O-F's strongest points is the cautious approach it takes to the user. This program doesn't know how to deal with double sided disks, but it doesn't just tell you so, it won't let you use them. You get a message clearly telling you that double sided disks are not-ok if you try. Similar messages appear if you try to use a disk that is flawed in a number of other ways.

Summary

O-F is available from SE MEDIA. It isn't really a program of general interest ... there are probably some OS-9 users who don't have FLEX or friends with FLEX. Those people have very little use for this program. The group of people this program should prove most useful to are the owners of software-switching machines. Using this program they can conveniently transfer data between

operating systems. There are a lot of FLEX users out there -- our close relatives in the computer world. It is good to be able to exchange disks with them even if we have to be the ones to provide the disks.

SE MEDIA
PO Box 794
Hixson, TN 37343
(615) 842-4601

- - -

A/D CONVERTER - JPC INTERFACE

WILLIAM A. NEELY, M .O
DEPARTMENT OF SURGERY
UNIVERSIT OF MISSISSIPPI MEDICAL CENTER
JACKSON, MISSISSIPPI 39216
601-987-5531

ANGEL K. MARKOV
DEPARTMENT OF MEDICINE
UNIVERSITY OF MISSISSIPPI MEDICAL CNETER
JACKSON, MISSISSIPPI 39216
601-987-5618

AUGUST 9, 1982

AN ANALOGUE TO DIGITAL CONVERTOR PATCH FOR MICROWARE SYSTEM CLOCK MODULE

INTRODUCTION

It became necessary in the course of interfacing a human calorimetry system to a Gimix 6809 System with 58K of memory and two small 5 1/4 inch discs to be able to convert eight channels of analogue data and one channel of digital pulse data for use by a basic program to perform certain calculations in order to compute oxygen consumption, carbon dioxide elimination and respiratory quotient. Also, weight, patient temperatures, two wet bulb and two dry bulb temperatures were n ded for the computations which were to be outputted each minute or a multiple thereof. The pulse input was needed to measure total airflow through the system using a dry test meter that closed t clock.l (or opened) a contact every liter of air flow. The details of the interface and the calorimetry system need not be discussed here except to point out these signals were conditioned to be in the range of 0 - 5 volts.

METHODS

A JPC Products Company AD - 16 analogue to digital conversion kit was purchased and assembled, tested with a basic program and worked entirely satisfactorily. A Motorola MC 6840 timer was used to measure the period of 100 pulses and thus enable the basic program to compute flow. To summarize, the requirements were to be able to convert eight channels of analogue data, store the results, move the results to a safe place for basic to read and convert the same eight channels one minute later (of course, this can be varied in the program if desired). Also the period of 100 pulses needed to be measured so flow could be determined.

RESULTS

```
0001                            XAM   Clock Module
0002                            IFPI
0004                            ENDC
0005                            Use   /d0/defs/systype
0006          * SYSTEM DEFINITIONS
0007  E050          A.IERM    EQU   $E050
0008  E220          CPORT     EQU   $E220
0009  0003          PPT       EQU   3
0010  0002          MC6840    EQU   2
0011  0001          M58167    EQU   1
0012  0001          CLKTYP    SET   M58167
0013  0002          MHZ2      EQU   2
0014  0001          MHZ1      EQU   1
0015  0002          CPUSPD    SET   MHZ2
0016  0000          END

0017          *****
0018          *
0019          * CLOCK MODULE
0020          *
0021          * INITIALIZE M58167 CLOCK CHIP FOR 100MS INTERVALS
0022          *    AND SETS IRQ POLLING ROUTINE
0023          *
0024          *****
0025          *
0026          * MODULE HEADER
0027          *
0028  0AC1          Type      SET   SYSTEM+OBJCT
0029  0002          Revs      SET   REENT+2
0030  0000 07C00221 ClkMod    Mod   ClkEnd,ClkNam,Type,Revs,ClkEnt,CPORT
0031  0000          ORG   0
0032  0000          BUFFER    RMB   56
0033  0038          BUFPTR    RMB   2
0034  003A          FMTPTR    RMB   2
0035  003C          COUNT     RMB   1
0036  003D          ADDATA    RMB   1
0037  003E          SAVFMT    RMB   2
0038  0040          SAFMAT    RMB   2
0039  0042          SECND     RMB   1
0040  0043          COUNTS    RMB   1
0041  0044          T1        RMB   2
0042  0046          T2        RMB   2
0043  0048          T3        RMB   2
0044  004A          TIMER     RMB   1
0045  0040          TIMER2    RMB   1
0046  004C          ENDBUF    EQU   .
0047  0000 436C6F63 ClkNam    FCS   /Clock/
0048  0012 03                 FCB   3           Edition number
0049  000B          CLKPRT    EQU   MSTAK       Stack has Clock Port address
0050          *
0051          * CLOCK DATA DEFINITIONS
0052          *
0053  0013 15       TINSVC    FCB   F8TIME
0054  0014 00F5               FDB   TIME-6-2
0055  0016 80                 FCB   $80
0056  0000          SecHilli  EQU   0
0057  0001          SecTenth  EQU   1
0058  0002          Second    EQU   2
0059  0003          Minute    EQU   3
0060  0004          Hour      EQU   4
0061  0005          DayWeek   EQU   5
0062  0006          DayMonth  EQU   6
0063  0007          Month     EQU   7
0064  0010          Status    EQU   16
0065  0011          Control   EQU   17
0066  0012          CountRst  EQU   18
0067  0013          LatchRst  EQU   19
0068  0014          RollOver  EQU   20
0069  0015          Spa       EQU   21
0070  E050          CAIREG    EQU   $E050
0071  E051          STATREG   EQU   $E051
0072  E052          ADDREG    EQU   $E052
0073  E053          CONTRG    EQU   $E053
0074  0000          TERMEN    EQU   $0000
0075  0000          CR13      EQU   0
0076  0001          CR2SR     EQU   1
0077  0002          MSBT1     EQU   2
0078  0003          T1LSB     EQU   3
0079  0004          MSBT2     EQU   4
0080  0005          T2LSB     EQU   5
0081  0006          MSBT3     EQU   6
0082  0007          T3LSB     EQU   7
0083  E210          FLOPRT    EQU   $E210
0084  0017 41444D45 MODNAM    FCS   /ADMEMORY/
0085  0011          MODTYP    SET   PRGRM+OBJCT
0086  001F 10       FORMAT    FCB   $10         CHANNEL 1
0087  0020 11                 FCB   $11         CHANNEL 2
0088  0021 12                 FCB   $12         CHANNEL 3
0089  0022 13                 FCB   $13         CHANNEL 4
0090  0023 14                 FCB   $14         CHANNEL 5
0091  0024 15                 FCB   $15         CHANNEL 6
0092  0025 16                 FCB   $16         CHANNEL 7
0093  0026 17                 FCB   $17         CHANNEL 8

0094  0027 27       ENDFMT    FCB   ENDFMT

0095          *****
0096          *
0097          * CLOCK INTERRUPT SERVICE ROUTINE
0098          *
0099  0028 6E9F0038 NOTCLK    JMP   [0.ISVC]    GO TO INTERRUPT SEAVICE

0100  002C 8EE210   CLKSRV    LDX   #FLOPRT     GET COUNTER ADDRESS
0101  002F A601               LDA   1,X         READ STATUS REG.
0102  0031 8504               BITA  #$04        IS IT COUNTER #3
0103  0033 2602               BNE   FLOIRQ
0104  0035 20ZE               BRA   CLKSRV1
0105  0037 FE0000   FLOIRQ    LDU   >TERMEN
0106  003A A601               LDA   1,X
0107  003C EC06               LDD   6,X         READ T3
0108  003E EDCB48             STD   T3,U
0109  0041 A601               LDA   1,X
0110  0043 EC04               LDD   4,X         READ T2
0111  0045 EDCB46             STD   T2,U        SAVE T2
0112  0048 A601               LDA   1,X
0113  004A EC02               LDD   2,X         READ T1
0114  004C EDCB44             STD   T1,U        SAVE T1
0115  004F 8683               LDA   #$83
0116  0051 A701               STA   1, X
0117  0053 8601               LDA   #$01        RESET TIMERS
0118  0055 A704               STA   0,X         TO BE SURE THAT
0119  0057 8600               LDA   #$00        ALL TIMERS ARE INITIAL.
0120  0059 A704               STA   0,X         BEFORE STARTING AGAIN
0121  005B AEB0FFAC           LDX   CLKPRT,PCR
0122  005F A68010             LDA   STATUS,X
0123  0062 2604               BNE   CLKSRV2
0124  0064 3B                 RTI
0125  0065 AEB0FFA2 CLKSRV1   LDX   CLKPRT,PCR GET CLOCK ADDRESS
0126  0069 A68010             LDA   Status,X    GET STATUS/CLEAR INTERRUPT
0127  006C 2704               BEQ   NOTCLK      BRANCH IF NOT CLOCK
```

```
W 005E FE0000   CLKSRV2 LDU   >ZERMEM    GO TO BUFFER SHIFT
  0071 8659             LDA   #$59       ON 59TH SECOND
  0073 A102             CMPA  Second,I
  0075 2707             BEQ   TICK1      PREVENTS GOING
  0077 C60A             LDB   #10        TO CLRBUF 10 TIMES
  0079 E7C843           STB   COUNTS,U   DURING 59TH SECOND
  007C 2007             BRA   TICK       JMPS TO CLRBUF
  007E 6AC843   TICK1   DEC   COUNTS,U   ONLY AFTER 10TH PASS
  0081 10270184         LBEQ  CLRBUF

  0085 4F       TICK    CLRA             SET DIRECT PAGE
  0086 1F8B             TFR   A,DP
  0088 170117           LBSR  ADCIRQ
W 008B 6E9FFFE0 TICK50  JMP   [CLOCK]    GO TO SYSTEM CLOCK ROUTINE



        *****
        *
        *  CLOCK INITIALIZATION ENTRY
        *
  008F 3408     ClkEnt  PSHS  DP         save Direct Page
  0091 4F               CLRA             clear DP
  0092 1F8B             TFR   A,DP
  0094 3401             PSHS  CC         save interrupt masks
  0096 860A             LDA   #10        SET TICKS / SECOND
  0098 975A             STA   D.TSEC
  009A 9759             STA   D.TIC
  009C 17008D           LBSR  ADINIT
  009F 8601             LDA   #1         SET TICKS / TIME-SLICE
  00A1 975B             STA   D.TSLC
  00A3 9748             STA   D.SLIC
  00A5 1A50             ORCC  #IRQM+FIRQM SET INTRPT MASKS
  00A7 308DFF81         LEAX  CLKSRV,PCR GET SERVICE ROUTINE
  00AB 9F32             STX   D.IRQ      SET INTERRUPT VECTOR
  00AD 327B             LEAS  -5,S       get scratch
  00AF 8E0054           LDX   #D.MNTH    Get Month ptr
  00B2 8D48             BSR   CNVBB      Convert Binary to BCD
  00B4 E7E4             STB   0,S        save month
  00B6 8D44             BSR   CNVBB      Convert
  00B8 E761             STB   1,S        save day
  00BA 8D40             BSR   CNVBB      Convert
  00BC E762             STB   2,S        save hour
  00BE 8D3C             BSR   CNVBB      Convert
  00C0 E763             STB   3,S        save minute
  00C2 8D38             BSR   CNVBB      Convert
  00C4 E764             STB   4,S        save second
  00C6 AE8DFF41         LDX   CLKPRT,PCR get clock address
  00CA CCFF02           LDD   #$FF02
  00CD A78813           STA   LatchRst,X Reset Latches
  00D0 A68810           LDA   Status,X   Clear any Interrupt
  00D3 E78811           STB   Control,X  enable 100 millisec line
  00D6 A6E4             LDA   0,S        retrieve Month
  00D8 2715             BEQ   SkipSet
  00DA A707             STA   Month,X    set Clock chip
  00DC A661             LDA   1,S        retrieve Day
  00DE 270F             BEQ   SkipSet
  00E0 A706             STA   DayMonth,X
  00E2 A662             LDA   2,S        retrieve Hour
  00E4 A704             STA   Hour,X     Set Clock Chip
  00E6 EC63             LDD   3,S        retrieve Minute & Second
  00E8 A703             STA   Minute,X   set Clock chip
  00EA 6F8815           CLR   So,X       reset seconds
  00ED E702             STB   Second,X
  00EF 3265     SkipSet LEAS  5,S        return scratch
  00F1 3501             PULS  CC         retrieve masks
  00F3 318DFF1C         LEAY  TIMSVC,PCR
  00F7 103F32           OS9   F$SSVC     SET TIME SEVICE ROUTINE
  00FA 3588             PULS  DP,PC

  00FC A680     CNVBB   LDA   ,X+        Get Binary byte
  00FE C6FA             LDB   #$FA       Init BCD byte
  0100 CB10     CNVB10  ADDB  #$10       Count Ten
  0102 800A             SUBA  #10        Is there a Ten?
  0104 24FA             BCC   CNVB10     Branch if so
  0106 5A       CNVB20  DECB             Count Unit
  0107 4C               INCA             Is there a Unit?
  0108 26FC             BNE   CNVB20     Branch if so
  010A 39               RTS
        *****
        *
        *  SUBROUTINE TIME
        *
        * RETURN TIME OF DAY
        *
  010C          TIME    EQU   *
  010C AE8DFEFC         LDX   CLKPRT,PCR GET CLOCK PORT ADDRESS
  010F 3401             PSHS  CC         SAVE MASKS
  0111 1A50             ORCC  #IRQM+FIRQM SET INTERRUPT MASKS
  0113 A602     TIME10  LDA   Second,X   get second
  0115 9758             STA   D.SEC      SET SECOND
  0117 A603             LDA   Minute,X   get minute
  0119 9757             STA   D.MIN      SET MINUTE
  011B A604             LDA   Hour,X     GET HOUR
  011D 9756             STA   D.HOUR     SET HOUR
  011F A606             LDA   DayMonth,X GET DAY
  0121 9755             STA   D.DAY      SET DAY
  0123 A607             LDA   Month,X    get month
  0125 9754             STA   D.MNTH     SET MONTH
  0127 A68814           LDA   RollOver,X CHECK FOR ROLLOVER
  012A 46               RORA
  012B 25E6             BCS   TIME10     BRANCH IF SO
  012D 3501             PULS  CC         RETRIEVE INTERRUPT MASKS
  012F 8E0054           LDX   #D.MNTH    GET DATE PTR
  0132 A684     TIME20  LDA   0,X        GET BCD BYTE
  0134 84F0             ANDA  #$F0       GET MSN
  0136 1F89             TFR   A,B        COPY IT
  0138 A884             EORA  0,X        GET LSN
  013A A784             STA   0,X        SAVE IT
  013C 54               LSRB             ADJUST MSN
  013D 54               LSRB
  013E 54               LSRB
  013F 54               LSRB
  0140 860A             LDA   #10
  0142 3D               MUL
  0143 EB84             ADDB  0,X        ADD LSN
  0145 E780             STB   ,X+        SAVE CONVERTED BYTE
  0147 8C0059           CMPX  #D.SEC+1
  014A 25E6             BCS   TIME20
  014C AE44             LDX   R$X,U      Get Specified location
  014E DC53             LDD   D.YEAR     Get Year & Month
  0150 ED84             STD   0,X
  0152 DC55             LDD   D.DAY      Get Day & Hour
  0154 ED02             STD   2,X
  0156 DC57             LDD   D.MIN      Get Minute & Second
  0158 ED04             STD   4,X
  015A 5F               CLRB             Clear Carry
  015B 39               RTS
        ******************************************************

        * SUBROUTINE TO INITIALIZE ADCONVERTOR
        ************************************************

  015C 308DFED7 ADINIT  LEAX  MODNAM,PCR ADDRESS OF MOD STRING TO FIND
  0160 8611             LDA   #MODTYP
  0162 103F00           OS9   F$LINK
M 0165 10BF0000         STY   >ZERMEM
  0169 8EE050           LDX   #DATREG
  016C 6F01             CLR   1,X
  016E 6F03             CLR   3,X
  0170 6F84             CLR   0,X
  0172 86FF             LDA   #$FF
  0174 A702             STA   2,X        ENABLE ADD INPUTS
  0176 8604             LDA   #$04       SET DONE ON...
  0178 A701             STA   1,X        CONV. COMPLETE
  017A 8634             LDA   #$34       RESET START CONV. LINE
  017C A703             STA   3,X
  017E 8EE210           LDX   #FLOPRT    GET FLOW COUNTER ADDRESS
  0181 CC0063           LDD   #99        NUMBER OF LITERS TO COUNT
  0184 ED06             STD   6,X        STORE IN COUNTER 3
  0186 CC3A98           LDD   #15000     FIRST NUMBER TO CASCADE
  0189 ED04             STD   4,X        STORE IN COUNTER 2
  018B ED02             STD   2,X        STORE IN COUNTER 1
        * PROGRAM THE CONTROL REG.
        * CONNECT PULSE TO C3
  018D 8600             LDA   #$00
  018F A701             STA   1,X
  0191 8640             LDA   #$40       SET CONTROL REG. 3 FOR CLK ,
  0193 A784             STA   0,X        INTERRUPT AT TIME OUT 6
        *                   EXTERNAL CLOCK
        * CONNECT 03 TO 62
  0195 8603             LDA   #$03       ALLOW ADD. OF CA1
  0197 A701             STA   1,X        ENABLE CLOCK,O ENABLE, INT. MS
        * CONNECT 02 TO C1
  0199 8601             LDA   #$01       ALLTIMERS PRESET C1 1$ INP., I
  019B A784             STA   0,X        EXTERNAL INPUT
  019D 8600             LDA   [illegible] STARTS TIMERS
  019F A784             STA   0,X        GO!!!!
  01A1 39               RTS

        **********************************************
        *   INTERRUPT ROUTINE FOR THE A-DCONVERTOR
        ************************************************
M 01A2 FE0000   ADCIRQ  LDU   >ZERMEM    GET START OF MEMORY
  01A5 30C4             LEAX  BUFFER,U   GET START OF BUFFER
  01A7 AFC838           STX   BUFPTR,U   SAVE FOR DEBUGGING
  01AA 318DFE71         LEAY  FORMAT,PCR FORMAT TABLE
  01AE 10AFC840         STY   SAFMAT,U   SAVE IT
  01B2 318DFE71         LEAY  ENDFMT,PCR END OF TABLE
  01B6 10AFC83E         STY   SAVFMT,U   SAVE IT
  01BA 10AEC840         LDY   SAFMAT,U
  01BE 10ACC83E ADREAD1 CMPY  SAVFMT,U   SEE IF DONE YET
  01C2 2713             BEQ   ADREAD2    IF SO EXIT
  01C4 10AFC83A         STY   FRMPTR,U
  01C8 8D0E             BSR   INPUT
  01CA 8D0C             BSR   INPUT      READ 1 CHAN.
  01CC 3121             LEAY  1,Y        INCREMENT FMT. PTR.
  01CE A7C830           STA   ADDATA,U   SAVE DATA TEMP.
  01D1 8D1D             BSR   SUMCONV    SUMMARIZE DATA
  01D3 3003             LEAX  3,X        INCREMENT BUF. PTR.
  01D5 20E7             BRA   ADREAD1
  01D7 39       ADREAD2 RTS
  01D8 A6A4     INPUT   LDA   0,Y        GET ADDRESS
W 01DA B7E052           STA   ADDREG
  01DD 863C             LDA   #$3C       START CONV.
W 01DF B7E053           STA   CONTRG
  01E2 8634             LDA   #$34
W 01E4 B7E053           STA   CONTRG
W 01E7 B6E051   INPUT1  LDA   STATREG    CHECK STATUS REG.

  01EA 2AFB             BPL   INPUT1     FOR CONV. DONE
W 01EC B6E050           LDA   DATREG     GET DATA...
  01EF 39               RTS               AND RETURN
        **********************************************
        *    SUBROUTINE TO SAVE TRIPLE PRECISION DATA
        ************************************************
  01F0 4F       SUMCONV CLRA             BE SURE A IS 0
  01F1 5F               CLRB              SAME FOR B
  01F2 E6C830           LDB   ADDATA,U   GET DATA FROM TEMP. STOR.
  01F5 EB02             ADDB  2,X        GET LOW BYTE & ADD TO DATA
  01F7 E702             STB   2,X        STORE IT IN ITS PLACE
```

```
00447  01F9 A981              ADCA  1,X       ADD CARRY AND MIDBYTE TO A
00448  01FB A701              STA   1,X       SAVE MIDBYTE
00449  01FD 2502              BCS   OVERFLO   SAVE CARRY
00450  01FF 2007              BRA   EXIT
00451  0201 49        OVERFLO ROLA            ROTATE CARRY TO BIT 0
00452  0202 9401              ANDA  #%00000001 SAVE CARRY ONLY
00453  0204 A884              ADDA  0,X       ADD HIBYTE TO CARRY
00454  0206 A784              STA   0,X       SAVE HI BYTE
00455  0208 39        EXIT    RTS
00456          ********************************************
00457          * SUBROUTINE TO TRANSFER BUFFERS TO SAFE
00458          * PLACE FOR BASIC TO READ AND ALSO
00459          * CLEAR ALL BUFFERS
00460          ********************************************
00461 M 0209 FE0000    CLRBUF  LDU   >ZERMEM   GET ADDRESS
00462  020C 6CC84A             INC   TIMER,U
00463  020F 6CC84B             INC   TIMER2,U
00464  0212 31C4               LEAY  BUFFER,U  ADD. OF START OF BUFFER
00465  0214 10AFC838           STY   BUFPTR,U  SAVE IT
00466  0218 C618               LDB   #24       SET UP COUNTER
00467  021A E7C83C             STB   COUNT,U   GET COUNT TO STOP ON
00468  021D A6A4      CONT    LDB   0,Y       GET BYTES
00469  021F 6FA4               CLR   0,Y
00470  0221 E7AB18             STB   27,Y      STORE BYTES AHEAD
00471  0224 3121               LEAY  1,Y
00472  0226 6AC83C             DEC   COUNT,U
00473  0229 26F2               BNE   CONT
00474  022B 16FE57             LBRA  TICK
00475  022E 37B700             EMOD
00476  0231           ClkEnd   EQU   *
00477                          opt   c
00479                          END
```

DISCUSSION

In order for the analogue to digital convertor and pulse timer to provide the data for a basic program memory space had to be allotted. This was done by giving the memory module a name (modnam) and type (modtype) as shown at decimal line 111. At line 370 the OS9 request F$LINK is used to return the memory address in the "Y" register. This address was next stored at memory location 0000 where it was accessible both to basic and any other programs use.

The M58167 clock interrupts every 100 milliseconds and even if there may be another interrupt the clock interrupt service routine is entered first. Immediately the MC6840 is checked to see if it is the source of the interrupt. If not the source the M58167 is checked. If it is not the source the routine branches back to the system. If the MC 6840 is the interrupt source, it is serviced, all timers read and values stored for later use. The RTI instruction at decimal 152 is to make sure the program returns to the regular interrupt routine, otherwise the system cannot find the other interrupts.

At decimal line 165 the CLKSRV2 routine prevents going to the clear and shift buffer routine ten times during the 59th second.

At line 379 the analogue to digital convertor is initialized as well as the MC6840 timer.

The routine beginning at line 410 reads the analogue to digital convertor and the routine at line 442 saves this data in triple precision format.

The routine of 461 moves these buffers down each minute and clears the original locations.

There is some redundancy in these routines but they work. It proved to be a painful debugging chore since there was no debugger to operate on top of the program.

The decimal numbers are not consecutive as many redundant assembly lines and pages have been deleted for publication.

BILL PHELPS AND DOUG ROBINSON OF MICROWARE HAVE AUTHORIZED THE PUBLICATION OF THEIR CLOCK MODULE WITH THIS PATCH

AAA SUPER MODEM

Over the past year or so we have seen an upswing in the demand for GOOD modem programs. Several have been published in 68 MICRO JOURNAL, however, they required an additional port or used an interrupt timer extensively, other than the terminal and modem port. While this in no way detracts from the operation of the system, once it is properly installed, it can be a nightmare to those who are not inclined towards the innerds of their computer. The AAA CHICAGO COMPUTER CENTER - 'Super Modem Program' does it all, and without interrupts or additional interfaces.

When we first got our BBS on line I needed a GOOD program to use at the home computer. There I act as SYSOP and also download files to our office. Jerry of AAA sent us his Super Modem Program to use on the home system, and it has performed like a champ!

General Overview

The modem program was originally written in 6800 machine language. It has since been redone for the 6809 with up-grades. It is furnished on either 5 or 8 inch disks and comes complete with source. To me personally this is a critical part of any software package. So, you not only get a very well documented instruction book but the source code (commented) as well. The source allows you to install it either on the FLEX™ or SSB DOS™ by changing a few bytes of code, for which the instructions are furnished. Special requirements for the GIMIX™ and other video boards are covered.

Operation

When the Super Modem Program is first loaded you are in local mode. This means that anything typed at the console is local and not transmitted beyond the console CRT. Data will still be received and displayed from the remote system in this mode.

All functions are menu driven. The menu can be recalled at any time, in case you forget what key fetches what function. Entering a control D will put you back into your DOS, or exit the modem program.

The following is a short review of each function:

1. control T - entry to the transmit function. A CR/LF is your prompt that you are actually in transmit. You will remain in the transmit mode until you enter an escape, at which time you are returned to local mode.

2. control F - this function will request (and not transmit any local interaction) the name of a disk file to be transmitted. You enter the file name, no extension, default to .TXT, and a CR, transfer of your disk file to the remote system is then accomplished. If you desire you may interleave keyboard activities with the file transmission, this would be useful for adding a note or other info to the transmitted file before it is closed. Therefore, you would first enter 'control T' then enter 'control F' to transmit a disk resident file from your system to the remote.

3. control E - should the remote system require an echo upon transmission, you would use this function to place the system in echo. Entering a second control E will toggle back. Control E may be entered at any time. This allows you to echo received data back to the sending system.

4. control R - this allows the reception, into your computer memory, a file from the remote system. nulls, deletes and linefeeds are ignored. Characters entered at the keyboard will be transmitted to the remote computer. Full duplex is supported without interrupts. When the file is completed, the control Q will save it to your disk system. It prompts for a file name if the sending computer does not send a 'auto-close'. Files larger than your available memory may be received for disk storage, if sending computer supports X ON/OFF.

5. control S - toggles full and half duplex.

6. control L - add a LF when a CR is received from modem. This is required by some systems that do not generate an automatic LF on receiving a CR.

7. control A - this function allows a delay to be inserted by your system should you have 'speed' problems. In our case we have our system come up in 'A' and also we

changed the default value to one more in line with our system. While this slows down the process somewhat, it allows us to transmit data to some systems that cannot receive characters as fast as we send them (300 baud). By turning 'A' on or off you have the luxury of working with systems that other modem programs will not work with. Provisions are made for additional delay, if required.

In the event you fail to receive the echoed character from the other system, or your system stalls, press any key and the transmissions can be restarted without loss of data.

The 'escape' key will terminate the current function and return to local mode and redisplay the command menu.

The 'X-ON and X-OFF' functions are supported and the 'ON or OFF' character may be changed to suit your particular needs. Again full source code is furnished for custom applications.

Control B - X-ON
Control C - X-OFF
Control V - start disk write
Control W - end of file

In the event control characters are received, which are not printable to the screen, the program indicates them with the underscore character, on the screen, with the exception of the backspace, line feed and carriage return.

One note that is worthy of special mention is the inclusion of a flow chart of the program. This allows source code modification much simpler. This is more a 'plus' than you might believe.

We have been using this program for over a year and have not found any flaws or bugs. Some rather large files have been sent and received using the Super Modem Program and *any* errors have been due to bad telephone lines (at our office). The nice part is that it uses no interrupts or additional hardware, other than another port for our modem. I have modified our version to come up in transmit and a few other things that I wanted for my particular needs. Because of the commented source code and the flow chart, these have been no problem at all. Most all users who dabble in machine code and using an assembler will have little or no problems. However, it works fine, right out of the box.

There are very few things needed to start up. You need to tell it which port you modem is located on and that is about all.

PRICE: **$75.00**

FROM:

AAA CHICAGO COMPUTER CENTER
120 Chestnut Lane
Wheeling, IL 60090
(312) 459-0450

COLOR User Notes

Robert L. Ney
5900 Cassandra Smith Rd.
Hixson, Tn. 37343

As usual, I'm running about 5 minutes before the deadline; this month just evaporated. I have received a lot of positive feedback on the last couple of months worth of listings; as I stated earlier, if you want the same thing for the Disk ROM, let me know. Obviously, those listings are just a beginning. What we need **NOW** is the info on those Subroutine Entries; Entry requirements, what it accomplishes, what Registers are affected and how, exit conditions, etc. We are hearing about a whole NEW set of ROMs for the Color Computer; Ver 1.2 BASIC, Ver 1.1 EXTENDED BASIC, and Ver 1.1 DISK BASIC that uses the full 8K ($C000 to $DFFF). Again, we will fill you in when we see what is happening.

If you are writing Assembly Language Programs for the Color Computer, remember that Radio Shack ONLY **GUARANTEES** their PUBLISHED Routine Entry Points; anything else you do may not work with a new Version ROM. They are better than nothing, but it sure would be a lot easier if they had followed some of the more meture DOS Procedures of defining firm Routine Entries for lower-level functions such as GetChar, PutChar, etc. (the 68xx standards of INCH, INCHE, INEEE, OUTEE, etc.). The High-Level functions Radio Shack provide severely restricts a programmers flexibility. The whole point is this; the Listings we just provided SHOULD be useable, especially where EXTENDED BASIC is using some of the Routines from BASIC, but they also could change, leaving a Program hanging "High and Dry". Just keep these points in mind. The only thing that SHOULD be FIRM is the Hardware Locations; if you are writing serious Software that you expect to be useable over an extended period, WRITE YOUR OWN I/O ROUTINES which interface DIRECTLY to the Hardware, and it SHOULD be safe. Maybe I am making too big a thing about this; it's just that some of us have become awful jittery when it comes to expecting any kind of consistency out of Radio Shack. Enough said!

— PROGRAMMING —

I thought I would begin a series of discussions concerning PROGRAMMING IN GENERAL. There are some excellent Books available which discuss "How to Program" with CERTAIN Languages, but very few which discuss Programming in General. This month we'll look at some OVERALL CONCEPTS and TERMINOLOGY, and get on into the meat of the thing later.

This presentation will not be directed specifically at any single Programming Language, although ANY attempt at presenting Efficient and Understandable Algorithms (big word; skip it for now if you don't understand it: I am just establishing, right up front, that I know and can use BIG Words, which indirectly leaves YOU, the impressionable READER, with the opinion that I know what I am talking about) invariably leads to a Pascal-Type of syntax and structure. Since that was the Overall Intent of Mr. Wirth and his Pascal, I suppose it is to be expected. Don't give up yet; I said GENERAL, which means this will apply to BASIC, PASCAL, C, ASSEMBLY LANGUAGE, etc. (even FORTH, to an extent).

OVERALL CONCEPTS:

Most new Programmers have the misconception that the actual Coding of a Program in the chosen Language is the HARD Part of Programming; THAT's the EASY PART of writing a Program. If the Program is properly DEFINED, it can be coded in almost ANY Language, and have a high probability of working correctly. My real complaint with BASIC, and the way it is normally presented, is that it induces the user to start right in writing Code, and requires an unacceptable amount of Debugging to get a Program running correctly. It is an extremely useful Language, and has a lot of power and capability, but it also tends to "Teach all of the WRONG Procedures" when it is the FIRST Language a user learns. If you need a quick, single shot, solution to a problem, BASIC is normally the way to go, because of it's flexibility and ease of use. I also want to state, before you start throwing rocks, that the BASIC Language provided with the Color Computer is one of the MOST POWERFUL BASIC's available on ANY Computer. It has enough of the Low-Level capabilities (such as DSKI$ and DSKO$) to allow you to do about anything you want with it. The Operating System is rudimentary, but the BASIC is very good (If we just had a ONERROR GOTO, we would really have something).

In general, a Program is DESIGNED, CODED, and TESTED. IF it is properly designed, the coding and testing is fairly straight forward. If it is a fairly large Program, subsections can be designed, coded, and tested, and then incorporated into the overall program. The overall Program, then, is built up of "Working Modules", and you should be fairly confident of its performance. This concept provides some side benefits. It is easy to change the Program by removing and/or replacing Modules with other Modules. You can also build up a file of

Modules that accomplish specific functions, such as Binary Search, Shell Sort, etc., which can be used in other Programs. Languages such as Pascal, C, etc., were designed to use this type of "Module" Structure. Assembly Language Programmers use modules in a "Library" File; while in BASIC they can be designed as Subroutines, or as a module that you would "Merge" or Insert into a Program while you are writing it.

If you haven't heard the term "Structured Programming", you are REALLY new to the Computer scene. Structured Programming has been batted around so much that the concept has become obscured by the details. Yet, Structured Programming is the salvation of the Software Writer. You can think of Structured Programming as you would of Outlining a Letter or Article; i.e., breaking the whole into parts, where each part is a Module. This Outline concept will also get you headed in the right direction in the "Top Down" problem solving methods, also. (Whoa; no rocks! I'm trying to present CONCEPTS, not precise terminology.)

Finally, you CAN NOT **EFFICIENTLY** design a Program without PROPER **DOCUMENTATION**! In other words, KEEP GOOD NOTES about what you are doing, how you are doing it, etc. When you change a part of the Program, keep the documentation up to date. This allows you to keep up with what was changed, why, when, etc. It also greatly simplifies debugging (If you listen to the experts, you should never NEED to DEBUG, right?).

With a concept of some of the "Parts and Pieces", we can now look at how to attack a Program. In general, it will go something like this:

1. Define the **DATA** Structures; i.e., what is going IN, and what is coming OUT. If the Program is a Game, the Data going IN may come from the Keyboard or Joystick, and the OUTput may be on the Display. If we are writing an Accounting Program, the INput Data will be in the form of Records, with the OUTput being Processed Records. If you think about it, there MUST be some Data to work on if we are even going to HAVE a Program.
2. Define **HOW** we are going to operate on this Data. This will be the different "Algorithms", or Procedures that we will use to accomplish different things. The "Top Down" approach to Problem Solving consists of breaking a large problem down into smaller and smaller parts, until we are at the point where the CODE to accomplish this part is **OBVIOUS** (that is why Coding is the EASY part). Some Programming Languages, such as FORTH, require that you define any Procedure BEFORE you use it, so IT ha to be Coded "Bottom UP"; but you STILL break the problem DOWN using the "Top DOWN" approach.
3. TEST the Program; i.e., run it and see where smoke appears. IF you have decent DOCUMENTATION (for reference, if needed), and have PROPERLY accomplished (??) the preceding two steps, testing should be as simple as finding some local idiot that has NEVER SEEN a Computer and see if he/she/or it can run/break the Program. Simple, huh?
4. Now for the HARD PART; **FINALIZE THE DOCUMENTATION**! If it is properly done, you can hand it to someone that knows the Language that the Program is written in, and THEY can **EASILY** add to your Program, debug your Program, etc. Remember, a month from now, YOU won't remember all of the details of the Program.

TERMINOLOGY

Some of the Terminology of Programming has already been mentioned, and you have a idea of what it means. Again, I am not going to attempt to be extremely PRECISE; I want to get the "Ideas" or "concepts" across, so that you can understand what you see when reading a Book on Programming, or on some specific Language, etc. Also, it will be easier to follow what we will be discussing later.

Top Down and Bottom Up:

I have already mentioned these concepts; lets look at them a little more. **TOP DOWN** is a "Logical Problem Solving" PROCESS! It is used in solving **ANY** Problem; whether it be repairing a piece of Electronic Equipment, writing a piece of Software, designing a Car, solving an Algebraic Problem, you name it. The procedure consists of breaking the WHOLE problem down into SMALL pieces, and breaking these down into STILL SMALLER pieces, etc. If I am going to repair a Transponder out of an Airplane, I first look at the Detected Video out of the Video Amp. If it is OK, I **KNOW** that the front half of the Radio, and most of the Power Supply, is operating normally (a Binary Search procedure). I have cut the Problem in half with ONE SIMPLE TEST! Solving a Programming Problem works the same way; break the overall Program down into smaller parts, and then break these down, etc., etc. It is easy, once you get used to it. Top Down PROGRAMMING, as opposed to Top Down PROBLEM SOLVING, means writing the OVERALL "Control" Program first, then different portions that it controls, etc. The easiest analogy is that of a Menu Driven Program; the Menu Program has overall Control over the WHOLE Program, and any part selected from the Menu is accomplished as called.

BOTTOM UP refers to building the actual Program from the lowest level back up to the top (FORGET IT as a Problem Solving approach). It is often called a "Building Block" process (although Building Block is NOT necessarily Bottom Up), from the similarity to building a house with "Building Blocks", or smaller pieces which make up larger ones, which are used to make up still larger ones, until the whole Program is written. As I stated, this is the procedure that is normally used with FORTH and Assembly Language Programs. Both can be programmed Top Down with the use of "Stubs" (later!), the method I personally prefer.

Procedures and Functions:

This is another area of confusion. Basically, a **PROCEDURE** is a method of controlling something, where a **FUNCTION** RETURNS A VALUE. Pascal has definite definitions and uses for each; other Languages "slide over the terms". Everything in C is a FUNCTION; BASIC has definite FUNCTIONS (SIN, COS, etc.); Assembly Language could care less. A lot depends on the specific literature you are reading as to the difference between the two. Again, I keep coming back to the "bare bones" idea that a PROCEDURE is a way of controlling program flow to accomplish some objective, while a FUNCTION **CAN** be the same thing, but NORMALLY returns some value.

Next month we'll continue the discussion and get into some examples. Till then;

BASIC FORUM (COCO)

James S. Upton
2631 Seventeenth Ave.
Greeley, Colorado 80631

Last month we examined a specialized word processing prograe. The prograe was originally written, in some haste, to provide a formatted output to an Epson MX-80 printer. Although the program was written for the classroom, it represented a type of prograe which is usable in the office, laboratory, home, or wherever a columnar representation of data is needed. Unfortunately the program suffered numerous flaws. You were challenged to become a program critic, and determine ways in which the prograe could be improved. A number of hints were given as to how you could direct these critical efforts, and an organized review of them would be appropriate at this time.

The intention of this critical process is to develop skills of self-criticism that will eventually lead to better programs of your own. The essential procedures consist of carefully examining all facets of the program, and critically questioning the results obtained. Several possible criteria were mentioned, and we may well review them too.

(1) A program should be convenient to use. By this I mean that the user of the program should be able to complete the intended task with no undue interference from the program itself. To be succinct, virtually any reasonably intelligent person should be able to use the program with ease. Things should not happen without adequate reason.

(2) A program should be efficient. It should operate with reasonable speed, and without unnecessary pauses. It should take advantage of the capabilities of the equipment which is being used.

(3) A program should be flexible. A program should do everything that is necessary, and should be capable of easily producing common variances. It should allow these things to occur easily, without confusion on the part of the operator.

(4) A program should be understandable. There are two meanings to this. It should make sense to the operator, and the programming code should be understandable to yourself, or another programmer, at a later time.

(5) A program should be functional. Let's face some facts here. I enjoy and appreciate the efforts of some of the graphics programmers. Indeed, a graphics program may demonstrate functionality. However it is sheer distraction to be in the midst of an accounting problem and have the screen suddenly covered with a beautiful color display. The program should fit the occasion and function.

These criteria are obviously not mutually exclusive. They overlap to a considerable degree. Perhaps equally obvious is that they may be applied to any programming language, and to virtually any programming circumstance. We will refer to them frequently.

Let's examine my promised improvements to the program. Example #1 is Version 2.0 of the formatting program. As you will see, the original program has been extensively modified. My critical examination of the original program found that none of the criteria had been met. Clearly something had to be changed.

Perhaps the single worst feature of Version 1.0 was the lack of convenience and flexibility. Imagine the position of the user who is entering four columns of twenty lines. Two columns and eleven lines into the input process the typist realizes, with some horror, that line eleven has just been entered twice! What options are available? Only two?! Neither of them are especially inviting. The operator may either continue, leaving in the error, or may reach for the <BREAK> key, and start over. Clearly a means of making a correction is needed.

Version 1.0 was organized around two loops, one for input, the other for output. Since I had decided to retain that relatively simple program structure, I could easily add a third loop which would allow reading through the text, and making corrections as necessary. Program lines 300 to 450 represent this effort. The delete and exchange procedure in line 400.

Version 1.0 was also lacking in efficiency and functionality. It should be clear that if one needs to have only four columns printed, 50% of the potential textual space has been lost. It is also noticable that while even numbers of columns will present a balanced appearance on the page, uneven columnar groups will appear assymetrically. This is simply not acceptable. As a last thought, it is highly inconvenient to keep "<ENTER>ing" for unused columns.

There is fortunately a means of simultaneously correcting both problems. Bring the entire process under control of the computer!

Version 1.0 formats through the capabilities of the printer. Printer formatting is convenient, but is neither flexible nor efficient. Can we find an algorhythm in BASIC that will take full advantage of page width, and allow balanced, flexible formatting? You may want to give this some thought before you continue with my solution.

Examine lines 40 to 110 of Example #1. This group is a "menu" which presents the user's first choice. It should be self-evident. Now look at lines 560 to 610, and lines 630 to 820. Here one will find two almost identical versions of the algorhythm that I finally decided to use. The use of numerous short subroutines was determined because that could correctly format both the headings, and the columns of text. The selected subroutine sets values for margin and text length. Lines 600 and 690 activate the shifts from one column to the next. Changing values of variable "K" set tab commands for the printer.

Now, if you have not typed the program into your CoCo yet, do it. Examine the BASIC coding as you type it in. Compare Version 1.0 to Version 2.0. Notice the changes in naming of variables. When possible, variable names have been related to their function in the program. Perhaps because my formal programming study was done with FORTRAN, I follow a convention that is sometimes associated with FORTRAN. Insofar as is possible I reserve the letters "I" through "N" for use with variables to pointers and counters. The original intention was that these letters were for Integer usage(as opposed to floating point). Some versions of BASIC allow this potential. The underlying intention is to increase speed in loops and the like. While it has no such function in the CoCo's Microsoft BASIC, it does help to organize your thinking, and to clarify program elements. I strongly recommend the practice.

As you enter, examine, and try out Version 2.0, you should continue using your critical powers. There are several improvements, "personalizations," or expansions which could be made. After you have examined and tested compare your ideas to mine which follow.

Even Version 2.0 contains minimal instructions to an operator. I have assumed that readers of BASIC FORUM will be avidly dissecting the program to see how it works, and what they can do with it. If such a program were put into the hands of a non-programmer, the instructions would be inadequate. A brief introduction could be placed ahead of the formatting menu. Perhaps a subroutine would be appropriate.

Both the input and review loops work their ways through a complete column, and then progress to the next. Some may prefer to enter text and/or review text by the line rather than the column. Give this some thought. If you do not see how this might be done, examine the output loop carefully. It is so organized. After due consideration I decided that I had no need for saving text, nor loading it. Others may prefer having this potential with either disk or tape. Either could be added through subroutines located at convenient places.

I am not especially fond of the appearance of the screen during the input and review processes. It is simple, and self-explanatory, but the overall appearance could be improved. Try this out.

The review and correct loop is possibly adequate when the text will never exceed one page. It is, however, cumbersome, and not especially efficient except in terms of the amount of memory consueed. You may want to try improvements here.

There are so many conceivable changes that could be made relating to the printer that I can only suggest a few possibilities. The most obvious is perhaps modifying the progrse to employ a different size of type. The modification would be extensive, but could be practical to many people. Converting to a different printer is also possible. One could not examine all printers, but we can look at one possibility, Radio Shack's new DMP-200.

Richard Matsumoto, who manages Radio Shack's computer operations in the Greeley area, graciously made it possible for me to use a CoCo and the new printer to adapt the formatter. This is not really a very difficult operation. Reverting to my "brute force" method. I loaded the progrse into the computer, and tried printing with the DMP-200. Only a few problems were found.

The DMP-200 sets type size through an external switch. CHR$(15) sets the underline function. Deleting line 480 corrects this problem. There are no italics available on this printer, so line 570 and the first half of line 620 may also be omitted. Since CHR$(172) in line 620 does not produce a desirable result, that line may be changed to: 620 PRINT @-2, STRING$(132,"="). The DMP-200 has no double printing, so lines 520 and 530 could also be omitted. There it is, an operational program for what appears to be a very good printer.

May I remind you that I welcome requests, suggestions, comments, and criticism. Next month we'll examine some questions of clarity and understandability in programming.

EXAMPLE NO. 1

```
10 CLS:PRINTTAB(7)"MX-80 FORMATTER"
20 PRINTTAB(11)"V. 2.0":CLEAR 7000
30 DIM TEXT$(12,56)
40 PRINT"SEVEN POSSIBLE FORMATS:"
50 PRINT"  (1) 2 COLUMNS OF 58 SPACES"
60 PRINT"  (2) 3 COLUMNS OF 40 SPACES"
70 PRINT"  (3) 4 COLUMNS OF 30 SPACES"
80 PRINT"  (4) 5 COLUMNS OF 24 SPACES"
90 PRINT"  (5) 6 COLUMNS OF 20 SPACES"
100 PRINT"  (6) 7 COLUMNS OF 17 SPACES"
110 PRINT"  (7) 8 COLUMNS OF 15 SPACES"
120 INPUT"ENTER THE NUMBER OF YOUR CHOICE";NUM
130 COL=NUM+1:WIDTH=INT(120/COL)
140 INPUT"HOW MANY LINES(TO 56)";LIN
150 LINEINPUT"ENTER TITLE: ";TITLE$
160 PRINT"ENTER COLUMN HEADINGS."
170 FOR I=1 TO COL
180 PRINTTAB(4)STRING$(WIDTH,"-")"<-COL. LIMIT"
190 PRINTI;".";:LINEINPUT HEAD$(I)
200 NEXT I
210 FOR I=1 TO COL:'INPUT LOOP
220 PRINT:PRINT"ENTER TEXT FOR COLUMN";I;".":POKE282,0
230 FOR J=1TO LIN
240 PRINTTAB(4)STRING$(WIDTH,"-")"<-COL. LIMIT"
250 PRINTJ;".";:LINEINPUT TEXT$(I,J)
260 NEXTJ,I
270 PRINT"DO YOU WISH TO REVIEW THE":POKE282,255
280 INPUT"TEXT (Y/N)";REV$
290 IF REV$="Y"THEN 300 ELSE 460
300 FOR I=1 TO COL:POKE282,0:'REVIEW LOOP
310 PRINT:PRINTTAB(5);"COLUMN #";I:PRINT
320 FOR J=1TO LIN
330 PRINTTAB(5);"LINE #";J
340 PRINTTEXT$(I,J)
350 PRINT"PRESS <ENTER>  TO CONTINUE"
360 INPUT"  OR 'C' TO CORRECT";CA$
370 IF CA$="C" THEN 380 ELSE 420
380 PRINT"ENTER YOUR REPLACEMENT."
390 INPUT"->";RP$
400 TEXT$(I,J)="";TEXT$(I,J)=RP$:RP$=""
410 PRINT:PRINT"CORRECTED->";TEXT$(I,J)
420 NEXTJ
430 NEXT I:POKE282,255
440 INPUT"RE-REVIEW OR PRINT (R/P)";Z$
450 IF Z$="R"THEN 300 ELSE 460
460 INPUT"IS THE PRINTER ON (Y/N)";PR$
470 IF PR$="Y"THEN 480 ELSE 460
480 PRINT@-2,CHR$(15)
490 INPUT"HOW MANY COPIES";CP
500 PRINT"DOUBLE SPACED (Y/N)"
510 INPUT"  (MAXIMUM OF 28 LINES)";SP$
520 INPUT"DOUBLE PRINTED (Y/N)";DP$
530 IF DP$="N"THEN 540  ELSE PRINT@-2,CHR$(27);"G"
540 J=LEN(TITLE$):K=INT((132-J)/2)
550 PRINT@-2,TAB(K)TITLE$
560 ON NUM GOSUB 760,770,780,790,800,810,820
570 PRINT@-2,CHR$(27);"4"
580 FOR I=1 TO COL
590 PRINT@-2,TAB(K)HEAD$(I);
600 K=K+L
610 NEXTI
620 PRINT@-2,CHR$(27);"5":PRINT@-2,STRING$(132,172)
630 FOR J=1 TO LIN:'OUTPUT LOOP
640 ON NUM GOSUB 760,770,780,790,800,810,820
650 IF SP$="N"THEN GOTO  660 ELSE PRINT@-2,""
660 FOR I=1 TO COL
670 IF I<COL THEN PRINT@-2,TAB(K)TEXT$(I,J);
680 IF I=COL THEN PRINT@-2,TAB(K)TEXT$(I,J)
690 K=K+L
700 NEXT I,J
710 M=M+1
720 IF M<CP THEN PRINT@-2,CHR$(12):GOTO540
750 END
760 K=6:L=60:RETURN
770 K=6:L=41:RETURN
780 K=5:L=30:RETURN
790 K=4:L=24:RETURN
800 K=4:L=20:RETURN
810 K=3:L=17:RETURN
820 K=3:L=15:RETURN
```

EXAMPLE NO. 2: Variable List

TEXT$	The text holding array.
NUM	Format pointer
COL	Pointer to number of columns.
WIDTH	Number of spaces in a column.
LIN	Number of lines to be printed.
TITLE$	As indicated.
HEAD$	The array holding column sub-titles.
REV$	Selection, review or print.
CA$	Correct or not.
RP$	Temporarily holds the replacement.
Z$	Choice, review or print.
PR$	Printer warning.
CP	Pointer to number of copies.
SP$	Double spacing pointer.
DP$	Double print pointer.
I & J	Loop pointers
K,L, & M	Formatting variables

HELIX DM64 MEMORY SYS

In the beginning, a few years past, here in the microcomputer business, there were the bad (dynamic) and good (static) guys, of memory. Or so they said. But then there was also a lot to be learned. We learned. And the result is that the bad guys wear white hats just like the good guys. Fact is you really can't tell them apart, that is if they are fed and treated as their maker decreed.

That was really part of the problem in the early days, too many designers did not understand and use them as they should. Like any other micro-device, they have their own special way of doing things. Static was easier to tame, but cost more, also they consume more power. Dynamic was less costly but require a specific diet. Mainly consisting of proper 'refresh' pulses, dosed out at the proper time. Once the refresh pulses were fed at the right time and the other particulars attended to, well, they (ynamic) looked, for all the world, like static. Dumb old CPU does not even know the difference. Just as well, it has enough to do already, especially as processor speeds continue to climb. Now comes a new breed of memory boards that run at the normal speeds as well and efficiently as their static cousins. They actually have not changed that much. We have just learned to fed them better.

The **HAZLEWOOD DM-64** dynamic memory board is one of the new breed. It is guaranteed to run at processo speeds in excess of 2 million cycles per second. Which is about as fast as any of our present processors clip along.

The board is fed 'refreshed' at 2 MHZ by using a "proprietary memory controller design." A newer technology than many existing schemes. The manufacturer claims that this system has better margins at 2.5 MHZ (pretty fast) than most existing dynamic memory boards running at a 1 MHZ rate. While we can't verify this to the 'dot', we have been running an early version, of this memory board, for over two (2) years and have not once experienced a memory failure or 'bomb-out'! Our system with the DM-64 has been very dependable.

Because of the particular manner in which the DM-64 is refreshed dynamic address translators (DATs) do not appear to affect the read access timing as in the case of some static boards.

Configuration

The DM-64 is is fully configurable by 4KB segments by on-board switches, and may be set for 64KB segments in a 1 megabyte (1,000,000) address range. Extended address lines may also be ignored by switch settings, allowing operation on systems where these lines are dedicated to other functions. Settup is ma e simple by these selection switches and allow you to configure to your system with a minimum of effort, skill and time. Adequate charts are furnished for configuration instructions. The DM-64 can be used in Standard S50 Bus systems as well as the newer 50/64 bus systems (HELIX).

Warranty and Construction

The board comes completely wired and tested to 2.5 MHZ operation. The warranty is one (1) year from date of purchase, by the USER. Some other boards are not always covered from the date of end-user installation. In this case they are, and that is an important consideration from those buying from a dealer who may have had the board in stock for a few weeks or even months. Just hang on to your sales ticket.

Experience

We have been running a DM-64 for over two years as I stated earlier. Our particular board has not had the regulator heat sink modifications. We have not experienced any memory problems. Later production boards have had few modifications from the original boards. The heat sinks on the regulators was one of the few updates and did eliminate some problems with the very first batch of boards. We are aware of users who had been running the DM-64 for over the one year warranty period, then began to experience or suspect that they were having memory problems. In every case that we are aware of, Hazlewood repaired or updated them at NO cost to the user, necessary or not. We consider that service.

Documentation

The instructions are adequate. The manual consist of a couple of sheets that basically cover what I have written above. Also is a full page configuration chart. The diagrams are complete and include a component layout chart. Despite the rather short manual, it is more than adequate for installation. The only thing missing is a 'blow by blow' description of what all happens (signal wise) on the board. While not necessary for installation, fuller signal descriptions do come in handy for a few of us 'do it yourselfers'. However, I guess we are a dying breed as most newer users are more concerned with out-of-the-box operation. Also if the time comes they will most likely want to ship it back to the manufacturer, or dealer for service. Actually that is a much better way. I have seen many good boards ruined by 'expert repairers'. Of course if yours works as well as ours has, there is no actual need for a lot of technical data.

The board is quality glass epoxy, double sided and silk-screened, with **gold edge connectors.** All ics are socketed. It is installed on the S50 bus and fits all popular cabinets.

We have recently installed the big-brother 256KB Hazlewood memory board in one of our 'use every day' computers and will have a report on it at a later date. It also is guaranteed to 2.5 MHZ.

For additional information contact:

HAZLEWOOD COMPUTER SYSTEMS
907 E. Terra
O'Fallon, Missouri 63366
(314) 281-1055

- - -

WESTCHESTER UTILITIES

Westchester Applied Business Systems FLEX9 Utilities

By: Steven M. Ward
39 Thorndike St.
Arlington, MA 02174

Introduction

There are many FLEX9 operating system utilities available for purchase from a variety of sources these days. I purchased several such utility programs from Westchester Applied Business Systems, P.O. Box 187, Briarcliff Manor, New York 10510. I have been using the utilities for over six months on an almost daily basis. The utilities I purchased include:

1) LISTDS displays the disk directory
2) MCOPY a single-drive file copy program
3) PMAP displays a binary file load map
4) DMAP displays a file sector map
5) DUMP displays memory dump in HEX/ASCII
6) DDSK displays disk sector in HEX/ASCII
7) ZAP disk sector binary editor
8) LOGO prints enlarged text
9) RESIZE adjusts text file line width
10) TYPESET justifies text file line margins
11) LISTM columnizes a text file

I will review each utility program in turn, but first I would like to to ch on several important areas of any software purchase.

Documentation

The documentation is quite good. Each utility has a single sheet of documentation which is printed so as to easily integrate with the TSC FLEX9 user's manual. Each page gives a brief description and includes a definition of the general command syntax and also includes several examples.

Support

When I received my disk and documentation I began to use my new software quite heavily. I discovered that a page of documentation for one of the utilities was in error. I also uncovered a "bug" in one utility and discovered that yet another utility didn't have all the features I considered essential to that utility program. While it is reasonable to expect correction of the first two problems, it is stretching "support" just a little on the third item. I should also point out that these utilities cost less then five dollars each, and though good product support should be provided by a vendor for all of his products, I was very surprised to receive outstanding service on all three of my problems.

The proprietor, Mr. Bill Adams, was pleasant, courteous, and prompt in his service. He sent me an amended documentation page along with the corrected bug and my suggested addition to the one utility program. Even though there were these initial problems, the excellent service more than compensated for them. The importance of good vendor support just cannot be overemphasized. Now on to each of the utility programs!

LISTDS

This utility displays the disk directory in three columns and all three columns fit within an eighty character display. The directory display consists of filename, starting and ending tracks and sectors, file size, and file creation date.

MCOPY

This is a single-drive file copy utility program. You can copy disk to disk with only one disk drive. MCOPY is loaded and then prompts for you to insert first the source disk and then the target disk so that MCOPY does not have to reside on either the file source or target disk. It copies a single file at one time as long as the source program does not exceed available memory size.

PMAP

This utility displays the memory load map and the file transfer address for a binary file. Each disk load record is displayed with it's starting and ending memory load addresses. The transfer address of the binary file is also displayed.

DMAP

This utility gives a disk map of a disk file by displaying a list of the file sectors in a track/sector format.

DUMP

This is a standard memory dump display program which displays memory simultaneously in HEX and ASCII.

DDSK

This utility gives a HEX/ASCII display of any disk sector with the address given as a track/sector disk address.

ZAP

This utility is a disk sector binary editor. It displays any sector in HEX/ASCII but extends several commands to the user to allow direct sector editing, if desired.

LOGO

This utility prints any word in an enlarged format of five print lines high by seven print columns wide per character. This allows giant program listing headers, etc.

RESIZE

The utilities, RESIZE, TYPESET, and LISTM comprise a very low cost, albeit simple, text processor. RESIZE will use a text file as input and generate an output file or listing with a new maximum line width. That is to say that you can input a file with say 80 columns and specify that all lines be readjusted to no more than 50 columns wide. Paragraphs will be honored.

TYPESET

After processing your file with RESIZE, now you can input the text to TYPESET where margins will be left and right justified. Naturally, you would normally specify the same line width that you specified for RESIZE. The drawback to this utility is that it does not honor paragraph indentation nor does it allow flagging tables, etc. for non-justification. Keeping tables in separate side files can get around this problem. This suite of text processing programs is very simple in function, but is also very simple to use. They are nice for the "light-duty" user, and the price is right! Paragraphs are honored by TYPESET.

LISTM

You can now input the text that was processed by TYPESET to LISTM where it will be formatted into the specified number of columns per page. Using these three programs together allows you to start with any text file and produce a margin-justified, multicolumn document quickly and easily. Certainly this is a very simple text processing package, but simplicity of operation does go with the simple functionality. Paragraphs are honored by LISTM. One nice feature is that these three utilities can be used separately. In fact, they must be used separately, one after the other to fully process an original text file.

Conclusion

I recommend these utilities to anyone as an excellent value. I also give high marks to Westchester Applied Business Systems for their product support, as well. Certainly in comparison to many utilities available offering similar features, these qualify as a "best buy".

EXTENED MEMORY TEST (CODE)

```
                              3-27-83 TSC ASSEMBLER   PAGE    1

                   * EXTENDED MEMORY TEST PROGRAM FOR SWTP DATRAM.
                   * PROGRAM ALTERS LOCATIONS $DFD0-$DFDF AND $FFF0-$FFFF
                   * ACCORDING TO EXTENDED MEMORY BOARD SELECTED BEFORE
                   * BRANCHING TO THE SBUG-E Q MEMORY DIAGNOSTIC ROUTINE.
                   * PROGRAM START ADDR IS AT $F400 AND REQUIRES A 6116
                   * 2K x 8 DRAM INSTALLED ON MP-09 CPU BOARD.
                   *
                   * WILLIAM A. COMER
                   * RT 1 BOX 437-C
                   * CATLETTSBURG, KY 41129
                   *                                   606-739-5438
                   *
             DFD0  DATRWTS  EQU   $DFD0     START ADDR OF DATRAM R-W TABLE

             DFDF  DAWRTE   EQU   $DFDF     END ADDR OF DATRAM R-W TABLE
             FFF0  DATACA   EQU   $FFF0     ADDR DAT RAM WRITE ONLY MEMORY

             F806  INCHE    EQU   $F806     GET/ECHO CHAR IN A-REG
             F80A  OUTCH    EQU   $F80A     OUTPUT CHAR A-REG
             F80C  PDATA    EQU   $F80C     OUTPUT STRING X-REG TO $04
             F9EF  SBUGO    EQU   $F9EF     ADDR SBUG-E MEMORY TEST
                   *
F 00                        ORG   $F400     START PROG EXECUTION ADDR
                   *
             F400  START    EQU   *
F400 8E    F555             LDX   #BDMSG    PROMPT FOR EXT BOARD NUMBER
F403 AD    9F F80C          JSR   [PDATA]   OUTPUT TO CRT
F407 AD    9F F806          JSR   [INCHE]   GET EXT BOARD NUMBER
F40B 81    30               CMPA  #$30
F40D 1027  00A0             LBEQ  Q0        IF BOARD 0 DO NOT MODIFY
F411 84    0F               ANDA  #$0F      CONVERT FROM ASCII TO DECIMAL

F413 C6    10               LDB   #$10
F415 3D                     MUL             10 X BOARD # IN B-REG
F416 F7    F5[illegible]    STB   EXBN      SAVE IN MEMORY
F419 BD    F57E             JSR   LFCR      LINE FEED CARRIAGE RETURN
                   *
                   * MEMORY TEST IN 0-48K  OR  48K - 56K RANGE
                   *
F41C 8E    F523             LDX   #RTMSG    POINT TO MESSAGE
F41F AD    9F F80C          JSR   [PDATA]   PROMPT FOR 0-48 OR 48-56
F423 AD    9F F806          JSR   [INCHE]   GET RESPONSE
F427 81    31               CMPA  #$31      1 = 48K-56K TEST
F429 27    22               BEQ   DOSRAM    SET UP TO TEST 48K-56K RAM
                   *
                   * SET UP FOR 0 - 48K RANGE TEST
                   *
F42B 8E    DFD0             LDX   #DATRWTS  POINT TO START OF R/W TABLE
F42E A6    84      LOOPT1   LDA   0,X       GET
F430 BA    F5[illegible]    ORA   EXBN      SET FOR EXT BOARD #
F433 A7    80               STA   0,X+      PUT
F435 8C    DFD0             CMPX  #DAWRTE   HALT ?
F438 26    F4               BNE   LOOPT1
F43A 8E    FFF0             LDX   #DATADR   POINT TO DAT RAM WRITE ONLY
F43D 86    0F               LDA   #$0F
F43F BA    F5[illegible]    ORA   EXBN

F442 A7    80      LOOPT2   STA   0,X+      *
F444 4A                     DECA
F445 8C    FFF0             CMPX  #$FFFB
F448 26    F8               BNE   LOOPT2
F44A 16    00A0             LBRA  Q
                   *
                   * SET UP TO TEST 48K-56K
                   *
F44D 8E    DFD0    DOSRAM   LDX   #DATRWTS  POINT TO START OF R/W TABLE
F450 86    03               LDA   #$03
F452 BA    F5[illegible]    ORA   EXBN      MODIFY FOR EXT BOARD #
F455 A7    80               STA   0,X+
F457 C6    02               LDB   #$02
F459 FA    F5[illegible]    ORB   EXBN      MODIFY FOR EXT BOARD #
F45C E7    84               STB   0,X
F45E 8E    FFF0             LDX   #DATADR   POINT TO DAT RAM WRITE ONLY
F461 A7    80               STA   0,X+
F463 E7    84               STB   0,X
F465 BD    F57E             JSR   LFCR
F468 8E    F472             LDX   #MSG
F46B AD    9F F80C          JSR   [PDATA]
XF46F 16   0078             LBRA  Q
                   *
F472 45 4E 54 45   MSG      FCC   /ENTER 0000-0FFF TO TEST C000-CFFF/
F476 52 20 30 30
F47A 30 30 20 30
F47E 46 46 46 20
F482 54 4F 20 54
F486 45 53 54 20
F48A 43 30 30 30
F48E 20 43 46 46
F492 46
F493 0A                     FCB   $0A
F494 0D                     FCB   $0D
F495 45 4E 54 45            FCC   /ENTER 1000-1FFF TO TEST D000-DFFF/
F499 52 20 31 30
F49D 30 30 20 31
F4A1 46 46 46 20
F4A5 54 4F 20 54
F4A9 45 53 54 20
F4AD 44 30 30 30
F4B1 20 44 46 46
F4B5 46
F4B6 0A                     FCB   $0A
F4B7 0D                     FCB   $0D
F4B8 04                     FCB   04
                   *
F4B9 BD    F57E    Q0       JSR   LFCR
F4BC 8E    F4CB             LDX   #B0MSG    POINT TO MESSAGE
F4BF AD    9F F80C          JSR   [PDATA]
F4C3 BD    F57E             JSR   LFCR
F4C6 20    25               BRA   Q
                   *
F4C8 41 4C 4C 4F   B0MSG    FCC   /ALLOWABLE RANGE FOR BOARD 0 IS 0-CFFF/
```

```
F4CC 57 41 42 45
F4D0 20 52 41 4E
F4D4 47 45 20 46
F4D8 4F 52 20 42
F4DC 4F 41 52 44
F4E0 20 30 20 49
F4E4 53 20 30 2D
F4E8 44 46 37 46
F4EC 04                 FCB     $04         END OF STRING POINTER
                        *
                        * GO TO SBUG-E MONITOR MEMORY TEST ROUTINE
                        *
F4ED BD   F57E          Q       JSR     LFCR
F4F0 8E   F4FA                  LDX     #ADDMSG     ISSUE PROMPT
F4F3 AD   9F F80C               JSR     [PDATA]
F4F7 7E   F9EF                  JMP     SBUGQ       GO TO SBUG-E MEMORY TEST
                        *
                        *
F4FA 45 4E 54 45        ADDMSG  FCC     /ENTER MEMORY TEST : START ADDR-END ADDR
F4FE 52 20 40 45
F502 40 4F 52 59
F506 20 54 45 53
F50A 54 20 3A 20
F50E 53 54 41 52
F512 54 20 41 44
F516 44 52 20 45
F51A 4E 44 20 41
F51E 44 44 52 20
F522 04                 FCB     $04
F523 45 4E 54 45        MTRMSG  FCC     /ENTER 0 TO TEST 0-48K ; ENTER 1 TO TEST
48K-56K :/
F527 52 20 30 20
F52B 54 4F 20 54
F52F 45 53 54 20
F533 30 20 34 38
F537 48 20 38 20
F53B 45 4E 54 45
F53F 52 20 31 20
F543 54 4F 20 54
F547 45 53 54 20
F54B 34 38 4B 20
F54F 35 36 4B 20
F553 3A
F554 04                 FCB     $04         END OF STRING POINTER
F555 45 4E 54 45        BNMSG   FCC     /ENTER EXTENDED MEMORY BOARD # TO TEST :
F559 52 20 45 58
F55D 54 45 4E 44
F561 45 44 20 40
F565 45 40 4F 52
F569 59 20 42 4F
F56D 41 52 44 20
F571 23 20 54 4F
F575 20 54 45 53
F579 54 20 3A 20
F57D 04                 FCB     $04         END OF STRING POINTER
                        *
                        * LINE FEED OR ROUTINE
                        *
F57E 86   0A            LFCR    LDA     #$0A        LINE FEED
F580 AD   9F F80A               JSR     [OUTCH]
F584 86   0D                    LDA     #$0D        CR
F586 AD   9F F80A               JSR     [OUTCH]
F58A 39                         RTS
                        *
                        * RAM LOCATIONS
                        *
                F588    ENDRM   EQU     *           RESERVE RAM MEMORY WORD
                                END

0 ERROR(S) DETECTED


SYMBOL TABLE:

ADDMSG F4FA   BNMSG  F555   DATADR FFF8   DOSRAM F44D   DRRVTS DFD0
DRVRTE DFDF   ENDRM  F588   INCHE  F806   LFCR   F57E   LOOPT1 F42E
LOOPT2 F442   MSG    F472   MTRMSG F523   OUTCH  F80A   PDATA  F80C
Q      F4ED   QBMSG  F489   QBMSG  F4C8   SBUGQ  F9EF   START  F400
```

SORTED DIRECTORY

by

F. James Rohlf
Dept. of Ecology and Evolution
State Univ. of New York
Stony Brook, NY 11794

The following program (written in Lucicata Pascal 3.9) prints on a printer and/or displays on a terminal a 3 column directory of the files on the disk in drive 1. The entries are sorted alphabetically -- which greatly helps in finding a misplaced file. The multicolumn display makes it possible to have the information from many files on the screen at once. The sector addresses are not displayed since they are seldom of interest.
The printed report makes use of the condensed print option found on most dot matrix printers. This makes it possible for the report to be cut out and taped to the floppy disk envelop. I have found this greatly helps in organizing ones disks. Small changes may be needed in two of the procedures if a printer other than a PRISM80 is used (one has to substitute the appropriate printer control codes).
A sample printout is furnished below.

```
Send to printer? (y/n)
Directory ( 4/14/83)
      Disk: FJR-WORK, #   4, Created:  4/14/83

COPY     .CMD   4  7/19/82  MISC    .TXT  72  2/18/83  PRINNEW .PAS  24  4/14/83
OPTMICRO.TXT  14  2/ 1/83  NEWRRS  .TXT  33  2/19/83  PDIRNEW .BIN   9  4/12/83
JUNK     .OUT   0  4/14/83  PDIR    .CMD  31  3/ 4/83  TERMINAL.ASM  38 12/10/82
LADOLCE  .TXT  50  2/19/83  PDIR    .PAS  18  3/ 4/83  XREF    .PAS  75  4/14/83
MICRO68  .TXT   5  4/14/83  PDIRNEW .BAK  20  4/12/83

Files: 14, Sectors: 323, Free: 289
Do another disk? (y/n)
```

```
program pdir(input,output,lpt);

        { print or just display sorted disk directory
          Gives file name & extension, size, and date.
          Up to 57 entries can be display on a CRT at
          once.

          Printed output is < 5" wide so it can be cutout
          and taped to (or inserted in ) the floppy disk
          envelope.

          The program assumes LUCIDATA Pascal, FLEX9 (but
          only a few address constants need be changed for
          FLEX), and a PRISM80 printer (procedures normprt
          and condprt will need to be changed for other
          printers).  Note; LUCIDATA Pascal assumes that
          the PRINT.SYS print divers have already been
          loaded (if you have not yet printed anything
          you could type GET PRINT.SYS).

          F. James Rohlf, 1983, Dept. of Ecology & Evolution
          State Univ. of New York, Stony Brook, NY 11794
        }
const
  maxlen = 120; { maximum no. of directory entries }
  null = chr(0);
  minrows = 5;  { minimum no. rows in directory table }

type
  fnameT = array[1..8] of char;
  fextT = array[1..3] of char;


var
  fcb : integer; { address of the FCB }

    fcode : byte;                   { FLEX file control block }
    errstat : byte;                 { many of these entries are }
    actstat : byte;                 { not actually needed by this }
    drive : byte;                   { program but are furnished   }
    fnamel : byte;                  { for reference               }
    fname : array[1..7] of char;
    fext  : fextT;
    fattribute : byte;
    byte16 : byte;
    startadd,endadd : integer;
    fsize : integer;
    fsecmap : byte;
    byte24 : byte;
    month,day,year : byte;
    fcbLptr : integer;
    curpos,currecno : integer;
    dataindex, randindex : byte;
    namewkbuf : array[1..8] of char;
    track,sector : byte;
    startindx : byte;
    firstdeldir : array[1..3] of byte;
    scratch : array[1..8] of char;
    spaceflag : byte;
    secbuffer : array[1..256] of byte;{ end of FCB }

  lpt : file of char;
  ans : char;
  i,j,k,kk,nrows,diskno : integer;
  nfree,nsec : integer;
  print : boolean;

  tabname : array[1..maxlen] of alfa;
  tabext  : array[1..maxlen] of fextT;
  tabsize : array[1..maxlen] of integer;
  tabdate : array[1..maxlen,1..3] of byte;
  tabptr  : array[1..maxlen] of integer;
  tablen : integer;
```

```
  {$ADDRESS = $CC0E }
  mm,dd,yy : byte;              { today's date from FLEX }
  {$STACK }

    { ** some procedures that may have to be modified ** }

procedure fmscall(fcb:integer);   { ** assumes FLEX9 }
          external $D406;
function address (var fcode :byte): integer;
          external $174;

procedure normprt; { select normal print 6 lines/inch,
                           12 chars/inch }
begin  { ** assumes PRISM 80 printer }
  writeln(lpt,chr(30),chr(27),'B,8,$');
end;

procedure condprt; { select print 8 lines/inch,
                           16.8 chars/inch condensed mode }
begin { ** assumes PRISM 80 printer }
  write(lpt,chr(31),chr(27),'B,6,$');
end;

  { ** end of procedures that may need to be modified ** }

PROCEDURE SORT ( n : INTEGER ) ; { Simple bubble sort }
   VAR                            { This could be replaced }
      i,temp : integer;           { if you have many files. }
      done : boolean;

   BEGIN
      REPEAT                      { sort pointers to files }
         done := true;
         FOR i:= 1 to n-1 DO
           IF tabname[tabptr[i]] > tabname[tabptr[i+1] then
            begin
             temp:=tabptr[i];          { out or order, swop }
             tabptr[i]:=tabptr[i+1];
             tabptr[i+1]:=temp;
             done := false
           END;
      UNTIL done
   END;

procedure line;       { display a line --- }
var
  i,j : integer;

begin
  for i:= 1 to 3 do begin
    for j:= 1 to 25 do write(lpt,'-');
    if i<>3 then if print then write(lpt,'   ')
    else write(lpt,'  ');
  end;
  writeln(lpt);
end;

begin                  { MAIN PROGRAM }
  writeln;

  REPEAT
    write('Send to printer? (y/n)'); { default is "NO" }
    readln(ans);  writeln;
    if (ans='Y') or (ans='y') then begin
      assign(lpt,'P:'); { printer }
      print := true;
    end else begin
      assign(lpt,'C:'); { output to CRT }
      print := false;
    end;
    rewrite(lpt);

        {  The following calls FLEX system info }

    tablen := 0;
    fcb := address(fcode);
    drive := 1;              { assumes drive #1 only,
                               this could be changed of course }
    fcode := 16;             { open system info record }
    fmscall(fcb);
    fcode := 7;              { get system info record }
    fmscall(fcb);


    if print then normprt; { normal print mode }

    writeln(lpt,'                Directory (',
              mm:2,'/',dd:2,'/',yy:2,') ');
    diskno := fattribute*256+byte16;
    write(lpt,'     Disk: ',chr(fname1),fname,', #',diskno:4);
    write(lpt,', Created: ',fsecmap:2,'/',byte24:2,'/',month:2);
    writeln(lpt);

    if print then condprt; { condensed print mode }

    nfree := fsize;
    nsec := 0;
    line;

    fcode := 6;                  { open directory }
    fmscall(fcb);
    while errstat<>8 do begin    { get file names from directory
      fcode := 7;                { get info record }
      errstat := 0;
      fmscall(fcb);
      if (fname1>0) and (fname1<127) then begin
        tablen := tablen+1; tabptr[tablen] := tablen;
        tabname[tablen] := '          ';
        tabname[tablen,1] := chr(fname1); { valid file name? }
        for i:= 1 to 7 do
          if fname[i]<>null then
            tabname[tablen,i+1] := fname[i];
        tabext[tablen] := '   ';
        for i:= 1 to 3 do              { file extension }
          if ext[i]<>null then tabext[tablen,i] := ext[i];
        tabsize[tablen] := fsize;
        nsec := nsec+fsize;
        tabdate[tablen,1] := month;   { file creation date }
        tabdate[tablen,2] := day;
        tabdate[tablen,3] := year;

      end;
    end;

    sort(tablen);    { sort pointers to file names }

          { output directory in 3 columns if possible }

    nrows := (tablen+2) div 3;
    if nrows<=minrows then nrows := minrows;
    if nrows>tablen then nrows := tablen;

    for i:= 1 to nrows do begin
      kk := i;
      for k:= 1 to 3 do begin
        if kk<= tablen then begin
          j := tabptr[kk];
          write(lpt,tabname[j]:8,'.',tabext[j]:3,' ',
                    tabsize[j]:3,' ',tabdate[j,1]:2,'/',
                    tabdate[j,2]:2,'/',tabdate[j,3]:2);
        end;
        if k<>3 then
          if print then write(lpt,'   ')
          else write(lpt,'  ');
        kk := kk+nrows;
      end;
      writeln(lpt);
    end;
    line;
    if print then write(lpt,chr(30),chr(27),'B,7,$');
    write(lpt,'Files:',tablen:3,', Sectors:',nsec:4,
              ', Free:',nfree:4);
    writeln(lpt);
    if print then page(lpt);
    write('Do another disk? (y/n)');  { default is NO }
    readln(ans); writeln;
  UNTIL (ans<>'Y') and (ans<>'y');
end.
```

BIT Bucket

Wilbur N. Killebrew, Jr. Feb 1, 1983 Page 2
File: [illegible]MACROS.[illegible]

TSC's FLEX09 assembler is the first macro assembler I have used in over eighteen years of computer experience. That must seem incredible to those who work in data processing. Only one of the machines I have worked with is even still in production, the Honeywell TDC4500. I doubt that most of you have even heard of it. It is descended from the GE 4000 series and their ancestor, the GE 412 which dates from the late 50's. Unlike most of the early machines, these are relatively addressed. They use a twenty-four bit word, probably unique in the industry.

I am new to the micro field, having been involved only just over a year. I was very much interested in the Altair when it first appeared; when I examined the Intel instruction set, I put it aside in total disgust. The Intel instruction set is awful, but their mnemonics are even worse, if that is possible! Zilog improved the mnemonics, but I can never forgive them for perpetuating and popularizing such a truly horrible instruction set. This is likely preaching to the saved, but even though the 6809 has some features that could some improvement, it is a very good programming machine. The 6809 instruction set is why, after all these years, I finally bought my own computer.

No doubt there are a lot of people who are becoming interested in assembly language programming but who, like me, haven't learned to use the power of macros. A continuing exchange of programming tools would be welcome. I could certainly use some good ideas to improve my own productivity.

There appears to be a fair number of 68XX owners who do not own assemblers with macro and/or conditional assembly capabilities. If you are one of these and are serious about writing in assembly language, you should obtain a good macro assembler with conditional assembly features. It will be well worth the money. Many feel they can't afford a lot of software, having mortgaged the farm just to buy some hardware. If you have Smoke Signal's assembler, patches have been published to implement macros. If you have something else, I suppose the only alternative is to write your own macro assembler or figure out how to patch what you have. Good luck! A group project would be better; too many cooks may spoil the pudding, and committees may build camels instead of horses, but a variety of views can lead to better software. Howard Harkness has proposed establishing a 68XX software library. It is a good idea. Where do we start?

GOOD PROGRAMMING TOOLS
MACROS AND CONDITIONAL ASSEMBLY

Wilbur N. Killebrew, Jr.

H. L. Harkness' remarks in 68 Micro Journal and RUGAB documentation set me to thinking about how to implement "real counters" for FLEX macros. I had been wrestling with the problem of ensuring unique labels created by repeated use of macros having internal labels.

My problem arose originally in converting macros from a Z80 macro assembler, Venus, to FLEX09. I wanted to make the FLEX09 macros as powerful as possible so the 6809 code would be as good as I knew howto make it. Some of you may have seen the project in Dr. Dobb's Journal. The macros and a program written in them implement TINCMP, a precompiler for a compiler compiler.

Recently the flash of inspiration came. I'll not be disappointed if I'm telling some of you something you already know if this will encourage others to help make this a continuing column on programming tools. Someone always seems to have a good trick that is not well known. And I don't mean the kind that make for obscure programming either. Good ones.

The Solution: use conditional assembly in a macro. It is so simple I don't know why I didn't think of it sooner. To make unique labels for repeated macro invocations, such as:

```
ABCD MACRO ARGUMENT
     :
     MAKELABEL STATEMENT,REMARKS
     :
     ENDM
```

the second macro, MAKELABEL, can supply the needed counter function and the actual labels. It can even be shared by any number of macros etc. The form as shown, creating a complete, labelled assembler statement looks quite clumsy, so one may prefer to use labelled equates. My counter macro looks like:

```
COUNT0 SET 0                  Initialize counter
MAKELABEL MACRO 'statement','remarks'
          IF COUNT0 = 0
LBL00     '&1','&2'
          EXITM
          IF COUNT0 = 1
LBL01     '&1','&2'
          EXITM
          :
          :
          IF COUNT0 = 99
LBL99     '&1','&2'
          EXITM
          ERR Counter 0 value out of limits
COUNT0    SET COUNT0+1
          ENDM
```


GIMIX INC. 1337 WEST 37th PLACE • CHICAGO, ILLINOIS [illegible] • [illegible] • [illegible]

April 27, 1983

Don Williams
'68' Micro Journal
P.O. Box 849
5900 Cassandra Smith
Hixson, TN 37343

Dear Don;

Gimix versions of FLEX are produced by Gimix under license from TSC for use only with Gimix controllers. [illegible]

We have not licensed anyone to use or sell our proprietary utilities, routines, or drivers other than for use with our FLEX and hardware.

We are of course aware that our versions of FLEX have become more popular than our sales of controllers would indicate. [illegible] have the costs of licensing, [illegible] Therefore, immediately, we are lowering the price of Gimix FLEX to $30.00 when ordered with any Gimix controller.

We are also offering to your readers a Gimix system specially priced at $2595.00. This system includes: [illegible] FLEX. Drives are available, but not included in this price. Our #28 double density controller may be substituted for $100 more.

Very truly yours,
GIMIX, Inc.

By Richard Don
Richard Don, V.P.

RD/jn

MIDDLESEX POLYTECHNIC
PSYCHOLOGY LABORATORIES
ENFIELD, MIDDLESEX
EN3 4SF
ENGLAND
Telephone 01 804 8131

I heartily agree with Ronald Anderson's Editorial in the October 1982 issue of 68 Micro Journal.
I have faced the problem of a mixture of SWTPC 1MHz and 2MHz CPU boards. Any user who has the same problems should

consider the following, before tearing their hair out.
1) Have you tried to run a 2MHz CPU board with other boards containing 1MHz components?
I found that the Eproms used in our system work when installed on a 2MHz CPU board, although created on a 1MHz computer.
2) What is the effective frequency of data transfer through the ports?
On our system (see APRIL '82 issue of 68 Micro Journal) the 2MHz computers, all SWTPC S-09's,communicate through old style SWT MP-LA parallel interfaces, with 6820's installed, to a 1MHz central computer.

If 1MHz boards disappear from the market many users, especialy home users will find the cost of upgrading too high. The alternatives are to give up using 68XX microcomputers or to go into do it yourself interfacing with 6809 home colour computers.
As Ron says in his editorial the 68000 is a great processor. However, the cost in the UK (and probably the US) is only worthwhile if you can go all the way. This means hard disks with a suitable high grade (expensive) printer and tape or double sided disk drives as backup. Is the 68000 computer a microcomputer?
Like many others we are committed to the 6809, on the SS-50 bus, and cannot change if for no other reason than finance.
Lastly, I must add that all opinions expressed in this letter are personal.

Yours Sincerely,

BRIAN ROBERTS

GIMIX inc. 1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609 • (312) 927-5510 • TWX 910-221-4055

April 25, 1983

Larry Williams
'68' Micro Journal
P.O. Box 849
5900 Cassandra Smith
Hixson, TN 37343

Dear Larry:

Enclosed is our new ad.

I call your attention to our new [illegible] systems featuring our new GMX III CPU, GMX III OS-9, and Intelligent I/O Processor card.

This system, with a 19MB winchester drive, is priced at only 70 cents more than the system in our prior ads. It offers in addition to the new CPU and I/O cards, [illegible] card the plus BASIC 09 and [illegible].

Please note that in all our OS-9 systems we are now including the OS-9 Debugger, Editor, assembler, BASIC 09 and [illegible] under license from MICROWARE.

We expect all GIMIX dealers to be able to give full support and service to our users. Any user who does not believe he is getting this support from any of our dealers should contact Bob Phillips or me directly. We assure you that we will take immediate measures. We intend that the quality of service, by GIMIX and its dealers, be equal to the quality of our equipment.

Very truly yours,
GIMIX, Inc.

By
Richard Don, V.P.

PRESS RELEASE

INTELLICOM

(An INTELLIgent COMmunications Program)

INTELLICOM provides you with the capability of very intelligent computer to computer communications from both a terminal emulation and file transfer standpoint. INTELLICOM supports several file transfer protocols that facilitate the transfer of both binary and ASCII data and files. Since INTELLICOM is menu driven, it is a breeze to use and understand.

With INTELLICOM you will be able to communicate with the various data and timesharing services such as The Source and CompuServe. Additionally, since INTELLICOM supports the protocol used by virtually all remote CP/M systems around the world, all users can immediately begin to take advantage of the wealth of public domain software available on these systems. Current, or potential, users of CompuServe can transmit and receive both binary and ASCII data with full error detection and recovery.

INTELLICOM will operate in the following modes:

- Terminal (full duplex)
- Terminal with local echo (half duplex)
- Terminal with data capture (either half or full duplex)
- CompuServe Executive

INTELLICOM provides the following modes of file transfer:

- Buffered ASCII capture with Xon/Xoff handshaking
- Buffered RCPM block checksum using Christensen protocol
- Buffered CompuServe "A" block checksum
- Buffered CompuServe ASCII capture (PTR/PTP protocol)
- Buffered Standard Intel Hex formatting

The checksum protocols allow for the verification of data blocks transferred (assuming appropriate support on the host end). This feature will be of great value in those applications where data integrity is paramount. INTELLICOM's documentation includes a detailed description of all protocols used along with machine readable examples of host pseudo code that will greatly ease the task of implementing support for INTELLICOM on any existing in-house mainframe (IBM, Dec, Wang...).

Customized (load and go) versions of INTELLICOM are currently available for the: IBM Personal Computer (PCDOS 1.1), NorthStar Horizon, NorthStar Advantage, Digital VT180/18x.CS9 and Flex 9.The NorthStar and Dec machines require CP/M 2.2. All systems must be equiped with a minimum of 56kb of memory, at least one disk drive, and a modem or direct connection to another computer. Other configurations are subject to availability and a slight surcharge. INTellicom can be purchased from DOTC ASSOCIATES. Please see ad 68 Micro Journal.

Jim Bellomo

Tri-Star Corporation

January 3, 1983

Mr. Don Williams, Sr.
Publisher
"68" Micro Journal
3018 Hamil Road
P. O. Box 849
Hixson, TN 37343

Dear Don:

I am sorry I am late sending this information as I told you I would do when I talked to you on the phone some weeks ago. However, I have been extremely busy and it just popped into mind this morning.

This is a simplified method of copying files from the hard disk to floppy for back up where the entire disk needs to be copied using double sided/double density formatted floppies.

Print a listing of the files on the hard disk and with an adding machine, starting with the #1 file, total the file record lengths until reaching approximately 3300 sectors. Draw a line across the file list at this point. Starting with the next file, again total approximately 3300 sectors, draw a line. Continue this process until the hard disk has been broken up into approximately 3300 sectors.

Insert the first floppy and copy hard disk to floppy. As the copying proceeds, watch for the file just prior to the first line. When it starts to copy this file, tap the tab or escape key once and wait until the file is complete. At this point, the system will shut down. Remove the floppy and insert another floppy. Tap the tab or escape key once and the process will continue. Continue this process until all the data has been copied off.

The disk you have copied this way will not operate until you run PUTBOOT on the floppy. At this point you have a complete copy of all the working files. Hope this helps somebody.

Very truly yours,

Robert G. Van Houten

RGVH/ei

A couple of items from **Bob Rosen** of **SPECTRUM PROJECTS** and "Bulletin Board" fame! The first is some information on "the **New Color Computer ROMs**. It is a list of the Addresses that are different from the previous Versions. **NOTE** especially the comments on the new **Disk ROM**; that will cause some problems. Also, here is a short summary of the necessary changes for updating Version -D and -E Boards to **64K RAM**; there is a lot of confusion about the mods. Most of the suppliers of the Mods have a good set of instructions: including **DATA-COMP**. I added the 33 ohm resistor note; I have seen three MC6883 Chips blown when the Mod was accomplished without this resistor.

--- RLN ---

SPECTRUM PROJECTS
93-15 86 DRIVE
WOODHAVEN,N.Y.11421

Voice Line (212)441-2807
Data Lines (212)441-3755/3766

64K "D" Board Upgrade

1. Remove capacitors C61,C31,C64,C35,C67,C45,C70,and C48.
2. Move the jumper at the right of U1D to the 16K position and remove the jumper plug between U8 and U4.
3. Make the following cuts:
   +5V to pin 9 of the rams
   +12V to pin 8 of the rams
   -5V to pin 1 of the rams
4. Add the following jumpers:
   +5V to the rams pin 1
   +5V to the rams pin 8
   Pin 35 of U10 to pin 9 of the rams. → USE 33.Ω RESISTOR
   Pin 12 of U4 to pin 16 of U8.
5. Bend pins 4,5, and 6 of U29 and pin 5 of U11 up in the air.
6. Connect pin 6 of U29 to pin 8 of U29.
7. Connect pin 4 of U29 to pin 5 of U11.
8. Connect pin 5 of U29 to TP1.
9. Install 64K chips in sockets U20-U27.

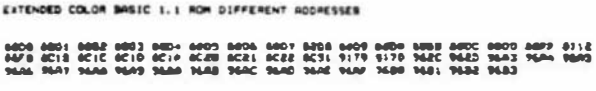
64K "E" Board Upgrade

1. Remove capacitors C61,C31,C64,C35,C67,C45,C70 and C48.
2. Set the jumper between U8 and U4 to the 32K position.
3. Set the jumper plug located just below C44 to the 16K/32K position.
4. Set each of the three jumper plugs located just above the keyboard connector to the 32K position.
5. Solder the two staking pins next to U29 to be in the LOW position.
6. Solder the two staking pins to the left of C44 together.
7. Bend pins 4,5, and 6 of U29 and pin 5 of U11 up in the air.
8. Connect pin 6 of U29 to pin 8 of U29.
9. Connect pin 4 of U29 to pin 5 of U11.
10. Connect pin 5 of U29 to TP1.
11. Install 64K chips in sockets U20-U27.

COCO BASIC ROM 1.2 ADDRESSING

ALL SOFTWARE WILL RUN ON THE NEW ROM WITHOUT PROBLEMS
THE ONLY CHANGES THAT I COULD FIND WERE SLIGHT CHANGES IN THE CHARACTER I/O
ROUTINES AND THE INTERPRET INTEGER EXPRESSION ROUTINE.

EXTENDED COLOR BASIC 1.1 ROM DIFFERENT ADDRESSES



ALL SOFTWARE WILL RUN CORRECTLY ON THIS ROM ALSO. THE ONLY DIFFERENCES I SAW WERE IN THE RCLEAR AND SOME OF THE GRAPHICS COMMANDS. THE RCLEAR HAS BEEN FIXED SO THAT IT NO LONGER GIVES AN ?SN ERROR WHEN YOU DO A RCLEAR AT THE BEGINNING OF A LARGE PROGRAM, OTHER THAN THAT, NO NOTICABLE CHANGES."

DISK BASIC 1.1 ROM DIFFERENT ROUTINES
THE DISK ROM HAS BEEN DRASTICALLY CHANGED AND MAY CAUSE SOME PROBLEMS AS FAR AS COMPATABILITY. A NEW COMMAND, DOS, HAS BEEN ADDED AND THE WHOLE ROM HAS BEEN MOVED UP TO MAKE ROOM. THE PROBLEM WITH THIS IS THAT ANY SOFTWARE WHICH EXPECTS DSKCON TO BE AT $D56C WILL GO THE WRONG ROUTINE, IT HAS BEEN MOVED TO ADDRESS $D75F. THE DOS COMMAND WHICH BOOTS IN AN ALTERNATE OPERATING SYSTEM SUCH AS OS-9 HAS BEEN ADDED AND IS REFERRENCED BY TYPING DOS, OR JSR [$C004]
THE ONLY OTHER CHANGE I COULD FIND WAS THE FIXING OF THE COPY COMMAND; INSTEAD OF LOCKING UP WHEN AN ERROR OCCURS AND THEN GIVING AN ?OS ERROR, IT RESPONDS WITH AN I/O ERROR AND EVERYTHING IS UNCHANGED. THIS NEW ROM CAUSES DEFINITE SOFTWARE INCOMPATABILITY PROBLEMS AND THOSE HAVING IT SHOULD BE WARNED THAT NOT ALL SOFTWARE WITH DISK ROUTINES WILL WORK.

COMPUTER SYSTEMS CONSULTANTS, INC.
1454 LATTA LANE
CONYERS, GEORGIA. 30207
TELEPHONE 404-483-1717 OR 4578

CSC OS/9 SLEUTH AND SIMULATOR IMPROVEMENT NOTICE NUMBER 4

The attached changes provide a correction which has been developed for the OS/9 SLEUTH, Z-80 SLEUTH, 6809 SIMULATOR, AND 6502 SIMULATOR programs. It affects the U command which allows the user to invoke shell functions. Please install these changes in the indicated modules. One or more of the changes may already be included in the modules. If you do not wish to install the changes yourself, return the original diskette for an update.

Thank You,

E. M. (Bud) Pass

```
******** xxxconst.txt ********
lon fcc " on"
 fcb $04
loff fcc " off"
 fcb $04
vshell fcs "shell"
ltext fcc "module "
 fcb $04
ldata fcc "storage"
 fcb $04
lxtnt fcc "extent "
******** xxxmiscl.txt ********
doxcom equ * call OS/9
 sts savas,u remember stack
 pshs x,y,u set up fork
 tfr x,u parameter area
 tfr x,y
doxpas tst ,y+ check for end
 bne doxpas
 ldd #$0d0d cr cr
 std -$01,y
 tfr y,d
 subd ,s
 tfr d,y parameter area size
 leax vshell,pcr "shell"
 clra language/type
 clrb size
 os9 f$fork spawn new task
 puls x,y,u
 bcs doxcer check for error
 pshs a save new id
doxcwt os9 f$wait wait for child task
 bcs doxcer
 cmpa ,s right task?
 bne doxcwt wait some more if not
 puls a restore child task id
 tstb check for child task error
 beq doxcox exit
doxcer lda #$02 error path
 os9 f$perr
doxcox lds saves,u recall stack
 rts
```

Dear Don:

I would like to give my impressions as a user of a new product recently introduced by CREATIVE MICRO SYSTEMS. The product, simply called the 9687 TABLETOP MICROCOMPUTER, features a hard disk, with 26 mbtes of formatted storage, a 96.00 3" half size floppy, and a 14 slot EXORbus motherboard. The power supply is of the switchmode design.

The microprocessor board, designated the 6809 runs at 2 mhertz. The on board features include a realtime clock, 48bp timer, two serials, and three Pio's. A 9678 host adapter (for the Hardisk) and a 9638 64K CMOS RAM module populated to 56K, fill two more slots leaving eleven open for expansion or specialization. Breakout boards provide the buffering between the CPU and the peripherals. The unit in teh minimum configuration, supports two serial terminals, and two parallel printers. A modem can be used in place of the second terminal if so desired. The rear panel has cutouts to accomodate eight additional serial devices, 2 additional drives and an EPROM programming head.

With memory management, and OS-9, level two, expansion to 1 mbyte of static memory is possible. This unit was delivered with OS-9 installed. Software included OS-9, level 1, Assembler, Editor, Debugger(disk) and Basic09. Also available for installation by the factory is the full complement of Microware products, the Stringraph Word Processor, Spelling Checker, and Mailmerge. Additionally, a Data Base Management program, and other languages can be installed at time of purchase or at a later date.

My principle use for the system is Word Processing. I am fully satisfied with the operation of stringraph under OS-9. It operates exactly as it does in my Flex System, with only minor variations on the command line and a much faster response. I have had only a smattering of experience with Basic09, but having transfered a program from sbasic to Basic09, I find that after adjusting for CPU speed, the transported program runs 3 1/2 times faster.

The transition from Flex to OS9 was about as can be expected. Software manuals 'assume' that you are proficient in the use of every program except the one your are using, as references to some interacting procedures could be made clearer. With use, the fog dissipates and the reasoning falls into place. Having used floppies for several years, the reasoning behind a multi-level directory became clear when I realized that it would be virtually impossible to locate a program on a 26 mbyte single level directory.

While the manuals for the software leave something to be desired, CREATIVE MICRO SYSTEMS has accomplished an outstanding job in the writing of their product specification. The appendices even include the dates for modification availability of software and hardware to move the CMS 9687 TABLETOP MICROCOMPUTER even more versatile.

Speaking of versatility, here is a partial list of the hardware support available for the CMS TABLETOP:

9638 128K Static Ram Module
also 64K, 32K Ram modules

9639 Memory Management Processor Module

9622-S Serial/Parallel I/O Module

Additionally, the CMS 9090 CRT Terminal and a Printer are available for the system. Since the basic hardware is designed to be adaptable, it is easy to see that a configuration for a specific requirement is easily assembled.

Some final thoughts. I was impressed at the auto-booting of OS-9. I am sure that this is not unique to the CMS 9687 system, but after punching in Flex at the start of each session, it was a pleasant surprise. Transfer of large programs from the hard disk are fast. Stringraph (about 25K) is on the line in about 2 seconds. The 9687 also supports the CMS 9614, 9617, and 6618 EPROM PROGRAMMING HEADS, if your needs include eprom burning. I have had th 9687 TABLETOP MICROCOMPUTER in operation for about 2 months without any problems. It is still a learning process for me. I find it difficult to go back to the old system after having had a taste of new Rolls Royce class of computers.

Sincerely,

Anthony J. Bessarre
Box 78
E.Sullivan, N.H. 03445
Ph (603) 847-9797

1313 N.E. Prescott
Portland, OR 97212
Phone (503) 284-2811
April 30, 1983

Dear Don,

Concerning TSC's FB.CMD, I made a few modifications that may be helpful. (I still use FLEX 2.0 and can't find justification to go the ASCO route.) Processing my thesis in chemistry there were several times that only the first few letters in a word needed to be underlined (to mean italics) and those letters were lower case.

First, the "CLSFT" subroutine called by " SET UNDEFLINED CHARACTER on page 16 of the manual needed to be moved so that lower case would be allowed. This new CLSFT was moved to $2200 to be out of the way and simply doesn't force upper case by subtracting $20.

Next, by using a DUMP AND REPAIR utility, or by modifying an EDITOR, set 130 (parity). I used BLITZ by MicroPal, since the source was handy, and strip after the letter to be underlined works well for me.

Strongely believe it; there are people using the 6800! and we appreciate your Journal.

Yours truly,

Gary Lemoine

Melkuist Ltd.
MICROPROCESSOR SYSTEM, DESIGN AND PRODUCTION
35A Guildford Street, Luton LU1 2NQ, Bedfordshire, England.
Telephone: 0582 416028 Telex: 825828 MELKST G

Your Ref:
Our Ref: SP/SM/4065
Date: 20th, January, 1983.

Computer Publishing Centre,
68, Micro Journal,
5900, Cassandra Smith,
P.O. Box 849,
Mixson,
IN37343,
U.S.A.

OS9 LEVEL 1 - FOR MOTOROLA EXORSET 30/33

Dear Sirs,

Melkuist Ltd. of Luton (U.K.) are pleased to announce a series of easily installed upgrades for the Motorola Exorset 30/33, which enable the the use of Microwares OS9. This implementation, unlike those before it uses standard OS9 diskette format, making the purchase and use of software from any source possible.

Please contact Sid Price at the address below for details or phone/telex on (0)582 416028, tex 825828 (U.K.).

Melkuist Ltd.,
35a, Guilford St.,
Luton, LU1 2NQ,
Beds.
U.K.

Yours faithfully,

S.Price
Director - Melkuist Ltd.

MD Mark Data Products

P R E S S R E L E A S E

Treat that itchy joystick finger to something special with the newest game release from Mark Data Products. GLAXIONS is a super hi-res space game for the Radio Shack Color Computer and TDP-100 that pits your playing skills against squadrons of swooping, diving enemy spacecraft. Your goal in this fast and furious game is to eliminate as many aliens as possible while avoiding your own destruction - not easy! Seven selectable skill levels coupled with automatic game acceleration provide a challenge for both novice and expert players.

This machine language program is available on 16K cassette for $24.95 and 32K disc for $29.95. GLAXIONS is available at your favorite dealer or may be ordered from Mark Data Products, 24001 Alicia Parkway, Suite 207, Mission Viejo, CA 92691.

24001 Alicia Pkwy. No. 226 • Mission Viejo, CA 92691 • (714) 768-1551

HI DON:

FIRST LET ME TELL YOU HOW MUCH I ENJOY YOUR MAGAZINE.

I HAVE BEEN A SUSCRIBER SINCE MID 79. I STILL USE THE BACK

ISSUES FOR REFERENCE MANY TIMES. ALSO WAS VERY PLEASED TO

SEE PETE STARK IN THE MAGAZINE TOO. HAVE FOLLOED HIM FOR

MANY YEARS.

I HAVE BEEN LOOKING FOR A COMPREHENSIVE BTTY PROGRAM FOR MY

6800 OR 6809 WITH DISK BUT TO NO AVAIL. MAYBE YOU CAN HELP

IF SO WOULD LIKE TO HEAR FROM YOU , OTHERWISE WILL KEEP LOOKING

AT THE MICRO-JOURNAL FOR IT. I HAVE A 69K SWTP BOX, 2 5 1/4 DRIVES.

THE COMPLETE BUSINESS SYSTEM

+Multiuser+Highly Expandable+Cost Effective

S+ THE CONCEPT

The S+ system is a modular computer system in which all portions of the hardware and software are designed to work together in the most efficient way possible. An S+ single user system with floppy disk storage is a competitive and cost effective entry level system. Unlike most other small computers being sold as "personal", or "small business" machines, the S+ system may be expanded to maximum capabilities using this same hardware and software. You cannot end up with a DEAD END system that cannot be expanded and whose software is not compatible with larger machines. A basic S+ system may be expanded to thirty-two users, a megabyte of main memory and hundreds of megabytes of hard disk storage by simply plugging in, or connecting the desired upgrade equipment.

TOTAL DESIGN—Hardware and Software

The S+ system is an integrated hardware and software design. The two complement and enhance each other in this system. The UniFLEX® operating system used in the S+ systems is patterned after the Bell Laboratories UNIX® operating system, one of the most admired and widely used operating systems in the world. Instead of being an afterthought, the software is part of the design of the S+ system. You can be sure that with this approach that all parts of the computer operate with maximum efficiency and cost effectiveness.

THE CENTRAL PROCESSOR

The basic S+ system is configured with 256K bytes of memory and can be expanded to more than 1 million bytes. An efficient and fast hardware memory management system is used to allocate the available memory among the users on a dynamic basis. As little as 8K bytes, or the entire memory—if needed—can be used by any individual user. This makes it possible to run very large programs on the system, but it also uses no more memory than necessary for a particular job. The increase in cost effectiveness of this system over crude and outdated bank switching arrangements is dramatic.

The central processor runs in both user and supervisor states. It can detect and reject a defective user program. It is impossible for a user program to go bad and stop the entire system, as can happen quite easily in less sophisticated systems.

Task switching is accomplished by use of a multiple map RAM memory, with sixty-four individual task maps. Each task can access from 4 to 64 K-bytes of memory. Multiple tasks may be used in programs that require more than 64K bytes of memory for execution. When a task is completed the memory is automatically released for other use.

SOFTWARE

The S+ operating system, UniFLEX® is a multiuser, multitasking operating system based on the UNIX® operating system that has been used for many years on Digital Equipment Corp. PDP-11 series minicomputers. It is considered one of the most sophisticated and "user friendly" operating systems available. Variations of UNIX® are rapidly becoming standard on mini and larger microcomputers.

A large variety of languages are available for use with the system. These include FORTRAN, COBOL, BASIC, and Pascal. Word processing packages are also available to give you full text processing capability on the system.

Applications programs are available in large quantities in many fields. This includes general business, medical, dental, veterinary, library and real estate management; plus others. Since the system is multiuser it can also be connected to cash registers to produce a point-ofsale terminal system combined with the computer. The possibilities for application of this system are endless.

THE I/O SYSTEM

The S+ system is totally interrupt driven. All terminal and printer I/O devices connect to an I/O bus separate from the main bus. Up to thirty-two separate devices may be connected to the I/O bus at any one time. If I/O activity is great enough to cause an unacceptable slowdown in system operation, a separate I/O processor can be installed in the system. This plug-in option removes all I/O handling overhead from the main processor and allows operation of up to thirty-two external devices at 9,600 baud. Without an integrated total design, as in the S+ system, it would become impractical to use a UNIX®type operating system in a situation with heavy terminal I/O activity.

DISK STORAGE

A wide range of disk storage capacity is available for the S+ system, from 2.5 M-byte floppy disks to an 80 M-byte Winchester and many sizes between. All disk controllers use direct memory access (DMA) type operations to maximize data transfer and to minimize overhead on the main processor. The Winchester disks also use intelligent controllers along with DMA transfers to preserve the performance that these type devices are capable of giving. Without this distributed intelligence the system performance would be greatly degraded. The UniFLEX®operating system is designed to work at maximum efficiency with this type disk system. The data transfer rates achieved by this combination rival those of large minicomputers.

COMMUNICATIONS

A high speed local network communications system is available to interconnect S+ systems. The VIABUS® network will allow communication between systems at data rates of over 400K baud. Such a system makes it possible to share data between local systems in an efficient and low-cost manner.

AVAILABLE SOON

Tape backup—20M-Byte in less than 15 minutes on a standard ¼ inch cartridge.

Mini-Wini—5 and 10 M-Byte Winchesters—5¼ inch package. Winchester performance, for smaller systems in a small package. UniFLEX® compatible design.

Large Capacity—190 and 340 M-Byte Winchesters, plus SMD cartridge drives.

UniFLEX is a registered trademark of Technical Systems Consultants, Inc.

UNIX is a registered trademark of Bell Labs.

VIABUS is a registered trademark of Southwest Technical Products Corporation.

PLEASE PUT THE FOLLOWING ITEMS IN YOU CLASSIFIED SECTION:

SWTP 6800 W/MP-B2,2-MP-C,3 MP-B,2 MP-LA,MP-T,MP-R, MICROTIME BD. AC-30 CASS.INT.,P&D MDI-1 DISK CONT. MUCH SOFTWARE $1150 TAKES ALL OR BEST OFFER 313-676-6856 DAVE BALKO 3005 CONCORD TRENTON, MI 48183

BEEHIVE 105 TERMINAL 24/80 19.2K VGC $250 U SHIP 313-676-6856 DAVE BALKO 3005 CONCORD TRENTON, MI 48183

THANKS DON. Dave F. Balko

DAVID F. BALKO
3005 CONCORD
TRENTON, MI 48183
(313) 676-6856

Dear Don,

I am writing to ask You or one of Your reader for VTL-09, a small BASIC-like language for the MC 6809 µP. VTL-09 is mentioned in "THE MC 6809 COOKBOOK" (TAB-Book 1209; p. 118) by Carl O. Warren.

Please could You tell me an address or reference, where I can get a version of this nice tool.

With best regards Klaus-Dieter Leyhr

Klaus-Dieter Leyhr
Roonstr. 17
0-7500 Karlsruhe 1
WEST GERMANY

I HAVE BOUGHT TWO VERSIONS OF FLEX FOR THE COCO. THE 'DATA-COMP' VERSION - F-MATE SEEMS TO WORK MUCH, MUCH BETTER. IT READS MORE DISK FORMATS (SWTPC, OIMIX, ETC) THAN THE OTHER, WITHOUT A LOT OF HASSLE. IT ALSO COMES WITH THE TSC ASSEMBLER AND EDITOR INCLUDED. THE SCREENS ARE MUCH EASIER TO USE AND READ. ALSO I HAVE HAD TO CALL BOTH, AND THE RESPONSE HAS BEEN GOOD AT DATA-COMP FOR F-MATE. THE OTHER I HAVE TALKED TO LADIES WHO DON'T KNOW MUCH ABOUT FLEX OR COMPUTERS, I STILL AM HAVING TROUBLE WITH THE FHL VERSION. HAVE NOT TRIED THE OTHERS, CAN NOT COMMENT.

IF YOU ARE LOOKING FOR A "GOOD" VERSION OF FLEX FOR THE COCO THEN I RECOMMEND THE DATA-COMP F-MATE. IT DOES NOT 'COST' ANY MORE AFTER YOU BUY THE 'EXTRAS' FOR THE OTHER...THE F-MATE IS COMPLETE!

BOB LASSINGER - LaGRENGE PARRISH, LA

P.S. AM LOOKING FOR A GOOD SOURCE FOR A CP/M TO FLEX PROGRAM THAT WILL RUN ON DATA-COMP F-MATE FLEX, ON MY COCO. I HAVE TWO DISK DRIVES (D0,D1). I HAVE A LOT OF CP/M SOFTWARE THAT I HOPE TO RUN ON THE 'MICRO-SOFT' BASIC IN MY COCO. I CAN GET IT ON A COCO DISK WITH THE PROGRAMS I GOT WITH MY DATA-COMP FLEX (RSDIR, RSREAD, RSWRITE), THEY WORK FINE. I JUST NEED SOME WAY TO GET THEM OFF MY 5" CP/M DISK. THANKS - WILL CHECK BACK IN EACH WEEKEND. ANOTHER P.S. - I SURE APPRECIATE FLEX OVER CP/M. LIKE GOING FROM A DARK ROOM, TO OUTDOORS INTO THE FULL SUN SHINE!!!!!!!!!!!!

PRODUCT RELEASE

For release after April 15

For additional information contact: Matt Astengo

NEW MICROPROCESSOR WORK STATION
INCLUDES EMULATOR AND
32 CHANNEL LOGIC ANALYZER

Bellevue, Washington, USA: The MODEL 4009 by Advanced Digital Technology combines an 8 bit Emulator with an analyzer-like 46 bit by 2048 line Trace, and an independent 32 channel by 1024 state Logic Analyzer, creating a single, easy to operate Microprocessor work station. These elements operate interactively or separately, under control of an independent microprocessor, with input and display provided by an operator-supplied video terminal. This combining of instruments eases and speeds the time consuming task of "integrating" microprocessor-based products.

ADVANCED EMULATOR PERFORMANCE

Newly developed emulation techniques provide bus timing characteristics that more closely match those of the microprocessor than was previously achievable. These techniques include:

1) Predictive clock generation by a proprietary tracking oscillator.
2) Stabilization of emulator timing through cooling of the processor.
3) Speed-up of time-critical signals by reactive loading.

Control of the Emulator is menu-orientated, enabling the operator to examine and store to memory, examine and alter registers, transfer data to and from Development Systems, test memory and busses, mask and generate reset and interrupts, trap on illegal instructions, begin emulation at any program point, run-to-breakpoint, and monitor clock and target system voltage. All microprocessor features are supported and the command menus may be used without effecting Emulator operation.

EXTENSIVE EMULATOR TRACE CAPABILITIES

A 46 bit by 2048 line Emulator Trace may be operated without interference of Emulation or may be set to stop emulation upon the occurrence of a sequence of events. 48 operator-defined events may be predefined and stored. Each event represents a pattern of 21 Address, 8 Data, Read/Write, 2 Bus- State, Illegal Instruction, 2 cycle-identifier and 4 Logic Qualifier bits, each definable as Logic 0, 1 or X (don't care).

Up to four events detected by hardware comparison may be used to specify a Trace Sequence, Restart Sequence and Trace Only Events. Each event may be passcounted up to 65535 times prior to detection. The trace is programmable to capture desired bus activity with selective exclusion of Bus-available cycles, DMA operations, invalid addressing (non-VMA) or irrelevant cycles.

An extensive choice of commands for the display of Trace, are available to perform comparisons of program activity with a previous trace record, to search for patterns or to disassemble program execution.

Single-step operation is also provided in trace display, or during register display.

LOGIC ANALYSIS FOR MULTI-PROCESSORS AND I/O

An optional 20MHz, 32 channel by 1024 state Logic Analyzer allows analysis of any desired points within the target system. Records are acquired either synchronously or asynchronously with respect to the Emulator and displayed independently or in a combined format with Emulator trace records. When displayed together, activity traced in both records is shown in order of real-time occurrence. The Logic Analyzer may be clocked from any of five internal sources, or from an external variable-threshold input. To provide cross-triggering, the passcounted events from the Emulator Trace are included in the Logic Analyzer event structure and events detected by the Logic Analyzer may become part of the event field of the Emulator Trace.

A MODULAR PACKAGE WITH FLEXIBLE I/O

The MODEL 4009's metal enclosure measures 17.5 by 15 by 5.75 inches and includes an Emulator card set and Pod assembly, Emulator Trace, Supervisory controller and one RS-232 port. Ordering options include Emulator card sets, Logic Analyzer card set and Pod Assemblies, 16K or 32K-Byte Memory Overlay, mapable in 512 Byte Segments, and optional I/O port cards which allow connection to a Host Computer, Prom Programmer and serial or parallel printers. 6809 and 68B09E processors are currently supported with a single Emulator card set with other microprocessor support to follow. Delivery 30-60 days ARO, Price $6500 and up depending on options.

ADVANCED DIGITAL TECHNOLOGY, INC., 13400 Northrup Way, Bldg. 27, Bellevue, Washington 98005 (206) 643-2382

```
50 REM  ( PRICE ) BY JAMES W. IVERS
60 REM                 4021 36TH ST.
70 REM                 DES MOINES, IOWA  50322
80 GOSUB 3500
90 OPEN "O,PRINT.CMD" AS 0
97 PRINT "A PROGRAM TO CALCULATE AND PRINT UP TO FOUR SELLING PRICES"
99 PRINT " FROM THE COST PRICE"
100 PRINT:PRINT
102 INPUT "HOW MANY PRICE COLUMNS DO YOU WANT?  1 TO 4 ";CX
104 IF CX=1 THEN LX=3
106 IF CX=2 THEN LX=2
108 IF CX=4 THEN LX=1
110 IF CX=1 THEN GOSUB 3720
112 IF CX=4 THEN GOSUB 3700
114 GOSUB 3500
116 PRINT "IS THIS ERROR RECOVERY ?  Y OR N ";:YN$=INCHR$(0)
118 PRINT
120 IF (N$<>"N" THEN 200
122 GOSUB 3500
124 GOSUB 3500
130 PRINT: INPUT "ENTER COMPANY NAME ";A$
140 GOSUB 3000
145 PRINT #0,TAB(65);DATE$
150 GOSUB 3000
155 PRINT
160 GOSUB 3020
170 GOSUB 3000
175 PRINT
180 GOSUB 3500
200 INPUT "ENTER PART NO....OR ESTOP] TO END ";P1$
205 PRINT:PRINT
210 IF P1$="STOP" THEN 1900
220 INPUT "ENTER DISCRIPTION ";P2$
225 PRINT:PRINT
230 INPUT "ENTER COST OF PODUCT";X
235 FOR J=L1 TO 4
240 GOSUB 3500
250 PRINT TAB(5);"1 - 20%";TAB(50);"6 - 33.3"
260 PRINT TAB(5);"2 - 22.2";TAB(50);"7 - 36%"
270 PRINT TAB(5);"3 - 25%";TAB(50);"8 - 40%"
280 PRINT TAB(5);"4 - 27.5%";TAB(50);"9 - 45%"
290 PRINT TAB(5);"5 - 30%";TAB(50);"10 - 50%"
345 PRINT:PRINT TAB(15);"11---WILL PRINT  (0.06)  HIT RETURN TWICE"
360 PRINT:PRINT
380 IF J=2 THEN 480
400 IF J=3 THEN 520
420 IF J=4 THEN 540
440 INPUT "ENTER CODE NUMBER  FOR G.T.C. (1 TO 11)";S
450 IF S >11 THEN 3600
460 ON S  GOTO 500,700,800,900,1000,1100,1200,1300,1400,1500,1600
480 INPUT "ENTER CODE NUMBER FOR 1 TO 5 ";S
490 IF S>11 THEN GOTO 3600
500 ON S GOTO 600,700,800,900,1000,1100,1200,1300,1400,1500,1600
520 INPUT "ENTER CODE NUMBER FOR  6 TO 11 ";S
530 IF S>11 THEN GOTO 3600
535 ON S GOTO 600,700,800,900,1000,1100,1200,1300,1400,1500,1600
540 INPUT "ENTER CODE NUMBER FOR 12 OR CTN. ";S
550 IF S>11 THEN GOTO 3600
560 ON S GOTO 600,700,800,900,1000,1100,1200,1300,1400,1500,1600
600 A=X/.8
610 A1=INT(A*100+.5)/100
620 IF J=1 THEN G1=A1
630 IF J=2 THEN H1=A1
640 IF J=3 THEN K2=A1
650 IF J=4 THEN K3=A1
670 NEXT J
680 GOTO 1700
700 B=X/.775
710 B1=INT(B*100+.5)/100
720 IF J=1 THEN G1=B1
730 IF J=2 THEN H1=B1
740 IF J=3 THEN K2=B1
750 IF J=4 THEN K3=B1
760 NEXT J
780 GOTO 1700
800 C=X/.75
810 C1=INT(C*100+.5)/100
820 IF J=1 THEN G1=C1
830 IF J=2 THEN H1=C1
840 IF J=3 THEN K2=C1
850 IF J=4 THEN K3=C1
860 NEXT J
880 GOTO 1700
900 D=X/.725
910 D1=INT(D*100+.5)/100
920 IF J=1 THEN G1=D1
930 IF J=2 THEN H1=D1
940 IF J=3 THEN K2=D1
950 IF J=4 THEN K3=D1
960 NEXT J
980 GOTO 1700
1000 E=X/.7
1010 E1=INT(E*100+.5)/100
1020 IF J=1 THEN G1=E1
1030 IF J=2 THEN H1=E1
1040 IF J=3 THEN K2=E1
1050 IF J=4 THEN K3=E1
1070 NEXT J
1080 GOTO 1700
1100 F=X/.9
1110 F1=INT(F*100+.5)/100
1120 IF J=1 THEN G1=F1
1130 IF J=2 THEN H1=F1
1140 IF J=3 THEN K2=F1
1150 IF J=4 THEN K3=F1
1170 NEXT J
1180 GOTO 1700
1200 G=X/.84
1210 G1=INT(G*100+.5)/100
1220 IF J=1 THEN G1=G1
1230 IF J=2 THEN H1=G1
1240 IF J=3 THEN K2=G1
1250 IF J=4 THEN K3=G1
1270 NEXT J
1280 GOTO 1700
1300 H=X/.8
1310 H1=INT(H*100+.5)/100
1320 IF J=1 THEN G1=H1
1330 IF J=2 THEN H1=H1
1340 IF J=3 THEN K2=H1
1350 IF J=4 THEN K3=H1
1370 NEXT J
1380 GOTO 1700
1400 I=X/.55
1410 I1=INT(I*100+.5)/100
1420 IF J=1 THEN G1=I1
1430 IF J=2 THEN H1=I1
1440 IF J=3 THEN K2=I1
1450 IF J=4 THEN K3=I1
1470 NEXT J
1480 GOTO 1700
1500 L=X/.47
1510 L1=INT(L*100+.5)/100
1520 IF J=1 THEN G1=L1
1530 IF J=2 THEN H1=L1
1540 IF J=3 THEN K2=L1
1550 IF J=4 THEN K3=L1
1570 NEXT J
1580 GOTO 1700
1600 N$=INCHR$(0)
1610 IF J=1 THEN G1=VAL(N$)
1630 IF J=2 THEN H1=VAL(N$)
1640 IF J=3 THEN K2=VAL(N$)
1650 IF J=4 THEN K3=VAL(N$)
1660 NEXT J
1680 GOTO 1700
1700 REM PRINT ROUTINES
1705 GOSUB 3500
1725 PRINT TAB(10);"COST AS ENTERED----";X
1730 PRINT:PRINT
1740 PF$="     \234567\ \Z3456789012345673901\  ###.##   ###.##   ###.##   ###.##"
1750 PRINT USING PF$,P1$,P2$,G1,H1,K2,K3
1760 PRINT:PRINT
1770 PRINT "ARE THE ENTRIES FOR THIS PART NO. CORRECT  (Y OR N)";:PR$=INCHR$(0)
1780 IF PR$()"Y"  THEN  190
1790 PRINT #0,USING PF$,P1$,P2$,G1,H1,K2,K3
1800 GOTO  190
1900 CLOSE 0
1910 END
2000 PRINT #0,TAB(40-.5*LEN(A$));A$
2010 PRINT #0:RETURN
2020 PRINT #0,"     PART NO.";TAB(14);"DISCRIPTION";
2040 PRINT #0,TAB(39);"G.T.C.";TAB(47);"1 TO 5";
2060 PRINT #0,TAB(55);"6 TO 11";TAB(64);"12 OR CTN."
2070 RETURN
3000 PRINT #0,TAB(5);
3005 FOR Y= 1 TO 69
3010 PRINT #0,"-";
3020 NEXT Y
3025 PRINT #0
3030 RETURN
3500 'PRINT CHR$(27);CHR$(69);CHR$(12)
3513 RETURN
3600 PRINT:PRINT:PRINT "CODE  NUMBER OUT OF RANGE---RE-ENTER."
3610 FOR Y=1 TO 1000
3620 NEXT Y
3630 GOTO 240
3700 PRINT:PRINT:PRINT  NUMBER OUT OF RANGE---RE-ENTER."
3710 FOR Y=1 TO 1000
3720 NEXT Y
3730 GOTO 120
```

WOODY BAKER
RT 11 BOX 4780
LUFKIN,TX 75901
(713)-639-3417

INFIX - POSTFIX EXP EVALUATOR

FORTH AND CERTAIN OTHER LANGUAGES USE REVERSE POLISH NOTATION DURING THEIR ARITHMETIC EVALUATION ROUTINES. INDEED, POSTFIX OR REVERSE POLISH NOTATION AS IT IS CALLED BY HEWLETT PACKARD, IS THE MOST EFFICIENT METHOD OF REPRESENTING ARITHMETIC EXPRESSIONS FOR A COMPUTER. ALMOST ALL COMPILERS CONVERT THE STANDARD ALGEBRAIC (INFIX) NOTATION TO POSTFIX BEFORE GENERATING MACHINE CODE FOR IT. THIS ROUTINE, WILL TAKE AN INFIX STRING CONTAINING AN EXPRESSION, AND CONVERT IT TO A POSTFIXED STRING. THE RESULTING STRING WILL HAVE $00 SEPARATING EACH OF IT'S ELEMENTS, INCLUDING VARIABLE NAMES AND OPERATIONS. THE LOCATIONS WITHIN THE PROGRAM THAT PRODUCE THE SEPARATORS ARE CLEARLY MARKED, SO THAT YOU CAN USE ANY KIND OF SEPARATOR THAT YOU WANT. FOR EXAMPLE, YOU MIGHT WANT TO US ,'S TO MAKE THE EXPRESSION COMPATIBLE WITH THE INPUT TO A MACRO ASSEMBLER, LIKE SSB'S MACRO ASSEMBLER. THIS ROUTINE USES A STACK TO PARSE THE EXPRESSION. IT DOES NOT ALLOW FOR LOGICAL OPERATIONS, THOUGH IT WOULD BE EASY TO MODIFY SO THAT IT DOES. LISTING 2 HAS THIS DONE. I SELECTED THE &=AND, !=OR AND ~=XOR. YOU CAN CHANGE THESE TO WHATEVER SYMBOL YOU WANT, WITHOUT MESSING UP THE OPERATION OF THE PROGRAM. I GAVE THE LOGICAL OPERATORS THE SAME PRIORITY AS ADDITION AND SUBTRACTION.

IT SHOULD ALSO BE POSSIBLE TO INCLUDE COMPARISON OPERATIONS ALSO. THE PROGRAM IS TREATED AS A SUBROUTINE, AND IS CALLED WITH X POINTING TO THE INFIX STRING, AND B HOLDING THE LENGTH OF THE STRING. IT RETURNS THE X REGISTER POINTING TO THE POSTFIX STRING. IT IS WRITTEN FOR THE 6800. IF I WERE WRITING IT FOR THE 6809, I WOULD USE THE USER STACK FOR THE PARSING STACK, THE X REGISTER FOR THE INFIX POINTER, AND THE Y REGISTER FOR THE POSTFIX POINTER. I WOULD APPRECIATE HEARING FROM ANYONE WHO USES THIS ROUTINE, AND IF ANYONE MAKES ANY IMPROVEMENTS TO IT, PLEASE SEND ME A COPY.

```
1.00=*****************************************************
2.00=* THIS ROUTINE TAKES AN INFIXED STRING AND CONVERTS IT TO A  *
3.00=* POSTFIX STRING. IT ALLOWS N CHAR VARIABLE NAMES, AND UP TO *
4.00=* TRIPPLE PRECISION CONSTANTS.  IT SUPPORTS *,/,+,- AND ()  *
5.00=* X MUST POINT TO THE ASCII STRING TO BE CONVERTED.  THE X  *
6.00=* REG. RETURNS POINTING TO THE ASCII STRING IN THE POST FIX *
7.00=* REGISTER B HOLDS NUMBER OF CHARACTERS IN INFIX STRING     *
8.00=*****************************************************
9.00=        ORG   $D100
10.00=DRIVER LDX   #IFSTR   ;INFIX STRING.
11.00=       LDA B #19
12.00=       JSR   PFIX
13.00=       JMP   $E000    ;GOTO MONITOR
14.00=IFSTR  FCC   'A=(B+C*2)/((G/D)-8)
15.00=*****************************************************
16.00=* INFIX TO POSTFIX CODE PRODUCER                      *
17.00=*****************************************************
18.00=PFIX   EQU   *   SET SPACING UP FOR THE LINE.
19.00=START  STX   IFXPTR  POINT TO INFIX STRING
20.00=       CLR   STKLEV  SET STACKLEVEL=0
21.00=       STA B IFXCNT  UP TO 255 CHARACTERS
22.00=       LDX   #PFIX1  POINT TO THE POST FIX LOCATION
23.00=       STX   PFIXPT  SET POINTER TO POST FIX STRING
24.00=PTOP   TST   IFCNT   SEE IF INFIX IS EMPTY
25.00=       BGT   NOTEMP
26.00=       LDX   PFIXPT  CLEAR STACK TO PFIX
```

```
27.00=CLEAR  PUL A
28.00=       STA A 0,X
29.00=       DEC   STKLEV
30.00=       BEQ   END      CLEAR ONWARD.
31.00=       INX
32.00=       BRA   CLEAR
33.00=END    LDX   PFIXPT   GET POINTER TO THE INFIX STRING
34.00=       RTS
35.00=*
36.00=*
37.00=NOTEMP LDX   IFXPTR   POINT TO THE INFIX STRING
38.00=NLOP   LDA A 0,X      GET A CHARACTER
39.00=       INX
40.00=       STX   IFXPTR
41.00=       CMP A #' '
42.00=       BEQ   NLOP     SKIP SPACES.
43.00=       LDX   PFIXPT   POINT TO WHERE TO STORE THE CHARACTER
44.00=       CMP A #'A'     TEST FOR VARIBLE
45.00=       BLT   SKIPCH
46.00=       CMP A #'Z'     TEST OTHER END
47.00=       BGT   SKIPCH   INVALID VARIABLE
48.00=LLOOP  STA A 0,X      MOVE VARIABLE NAME
49.00=       INX
50.00=       CLR   0,X      $00 IS SEPARATOR
51.00=       CLR   1,X
52.00=       STX   PFIXPTR  POINT TO THE INFIX STRING
53.00=       DEC   IFXCNT
54.00=       LDA A 0,X      KEEP ANAYLIZING STRING
55.00=       INX
56.00=       STX   IFXPTR   KEEP POINTERS UP TO DATE
57.00=       LDX   PFIXPT
58.00=       CMP A #'A'     SEE IF MULTI CHARCTER VARIABLE NAME
59.00=       BLT   SKIPC2
60.00=       CMP A #'Z'
61.00=       BGT   SKIPC2
62.00=       BRA   LLOOP    AS LONG AS ALPHA CHARACTERS, MOVE VAR.
63.00=*
64.00=*
65.00=SKIPC2 INX            LEAVE A SEPARATOR
66.00=SKIPCH STX   PFIXPT   SAVE POST FIX POINTER
67.00=       LDX   IFXPTR   POINT TO CORRECT TOKEN
68.00=       DEC   IFXCNT   REMOVED 1 FROM INFIXED STRING
69.00=       CMP A #'('     SEE IF PARENTHESIS
70.00=       BNE   NORP
71.00=PUSH   PSH A          IF IT IS THEN STACK IT
72.00=       INC   STKLEV
73.00=PTOP1  BRA   PTOP
74.00=PULL   PUL A          UNSTACK IT
75.00=       DEC   STKLEV
76.00=       BRA   PTOP
77.00=*
78.00=*
79.00=NORP   CMP A #')'  SEE IF OTHER
80.00=       BNE   SEMPTY   CHECK FOR STACK EMPTY
81.00=       BRA   OPLOOP
82.00=*
83.00=*
84.00=OPLOP1 PUL A          GET A
85.00=       DEC   STKLEV   KEEP UP WITH STACK LEVEL
86.00=       LDX   PFIXPT
87.00=       STA A 0,X
88.00=       INX
89.00=       STX   PFIXPT   SET IT BACK UP
90.00=OPLOOP PUL A
91.00=       PSH A          GET THE OPERATOR
92.00=       CMP A #'('
93.00=       BNE   OPLOP1   WIPE NEXT ONE
94.00=       PUL A
95.00=       DEC   STKLEV
96.00=       BRA   PTOP1
97.00=*
98.00=*
99.00=SEMPTY TST   STKLEV   CHECK STACK LEVEL
100.00=      BEQ   PUSH     IS EMPTY
101.00=      PUL A          GET TOS
102.00=      PSH A          FIND TOS
103.00=      CMP A #'('
104.00=      BEQ   PUSH2    A HOLDS THE VALUE..X=>OPERATOR
105.00=      CLR   PRIOR1
106.00=      CLR   PRIOR2
107.00=      BSR   PRIOR1   CHECK PRIORITY
108.00=      LDA A PRIOR2
109.00=      STA A PRIOR1   TAKE OPERATOR PRIORITY AND SAVE
110.00=      CLR   PRIOR2
111.00=      DEX
112.00=      LDA A 0,X    GET THE OPERATOR
113.00=      INX
114.00=      BSR   PRIOR1
115.00=      LDA B PRIOR1
116.00=      LDA A PRIOR2   GET THE PRIORITIES
117.00=      CBA            COMPARE
118.00=      BGT   PUSH2    IF 2>1 THEN PUSH 2
119.00=      PUL A
120.00=      DEC   STKLEV
121.00=      LDX   PFIXPT
122.00=      STA A 0,X
123.00=      INX
124.00=      STX   PRIXPT
125.00=      BRA   SEMPTY   STACK EMPTY
126.00=*
127.00=*
128.00=PUSH2 DEX            FIND PREVIOUS OPERATOR
129.00=      LDA   0,S
130.00=      INX
131.00=      BRA   PUSH
132.00=PRIOR1 CMP A #'+'
133.00=      BEQ   EXP
134.00=      CMP A #'-'
135.00=      BEQ   MULOP
136.00=      CMP A #'/'
137.00=      BEQ   MULOP
138.00=      CMP A #'*'
139.00=      BEQ   ADDOP    SECOND LOWEST PRIORITY
140.00=      CMP A #'='
141.00=      BEQ   ADDOP
142.00=      CMP A #'='
143.00=      BEQ   ASSGN
144.00=      BRA   RETURN   MAY WANT AN ERROR MESSAGE
145.00=*********************************************************
146.00=* CALCULATE PRIORITY ROUTINE.  NUMBER 4=HIGHEST 1=LOWEST  *
147.00=*
148.00=*********************************************************
149.00=EXP   INC   PRIOR2
150.00=MULOP INC   PRIOR2   SET PRIORITY UP
151.00=ADDOP INC   PRIOR2
152.00=ASSGN INC   PRIOR2
153.00=RETURN RTS           RETURN WITH PRIOR2=PRIORITY NUMBER
154.00=*********************************************************
155.00=* VARIBLE ALLOCATION BLOCK                              *
156.00=*********************************************************
157.00=STKLEV RMB  1
158.00=IFXCNT RMB  2
159.00=PFIX1  RMA 256
160.00=PFIXPT RMA 2
161.00=PRIOR1 RMA 1
162.00=IFXPTR RMB 2
163.00=PRIOR2 RMB 1
164.00=       END  DRIVER
```

COMPARE - 6809 - Z80

David Shearer
83 Marcella St.,
Scarborough, Ontario.
Canada. M1G 1L3.

Introduction

I would like to help set into perspective some of the facts, myths, and innuendo that surround the on-going comparison between the 6800 software world and the CP/M software world. This discussion on numerous occasions has become a good sure versus bad sure type of dialogue characterized by rather emotional "taking of sides". I, of course, will stand accused of harbouring subjective opinion (and correctly so!) but perhaps with a slight difference! I have implemented from the ground up and owned both a Flex based 6800 system and a CP/M based Z80 system of comparable configuration, cost, and capability. In the process of implementing and subsequently using both systems I have accumulated literally hundreds of hours of exposure to the internals of each. Professionally, I am employed as a technical manager with one of the "Big Five" computer companies and currently have responsibility for support of four major operating systems - two mini-computer based and two mainframe based, so hopefully my comments will be of use and of interest to the readers of '68' Micro Journal.

The comparisons rage on at several different levels: the actual processing elements 68XX vs. 8080/8085/Z80, the respective bus architectures 8-50 vs. S100, the operating systems Flex or OSA vs. CP/M, and the applications, languages, and utilities which are available for each. Each camp comprises a knowledgeable and relatively large body of users who are supported by their respective flagship publications - '68' Micro Journal for the 68XX folks and Byte magazine for the CP/M people. I subscribe and read every issue of both and frankly am more than happy with each.

Comparing the Microprocessors

Let's start with the chips but with one proviso: this is not going to be a hardware oriented discussion as I will defer to others much more qualified than I in that regard. In my opinion, from a register architecture and instruction set point of view, the 6809 is hands down far and away the most elegant processor in the 8 bit world. Although the 8080 and particularly the Z80 at first glance offer a wider range of registers, the 6809 easily compensates with the simple elegance of its instruction set and addressing modes. It is true that the Z80 offers a superset of the functionality of the 8080 or 8085 devices. But an often overlooked fact must be remembered. Seldom is this extra functionality used as people who write CP/M software do so for the largest possible target audience and therefor limit their instruction usage to 8080 instructions only. Using Z80 specific functions would preclude usage by CP/M users with 8080 or 8085 processors. The moral of the story is that the Z80's true potential is rarely exploited. Note that CP/M is officially shipped by Digital Research (its authors) for usage in an 8080 environment. Last but not least, the 6809 has a vast advantage in its ability to address memory beyond the 64K limit of the other 8 bit processors.

Both chips are manufactured by large and professional semiconductor manufacturers, and both processors are well supported with a family of companion chips such as input output controllers, DMA controllers etc. The quality and quantity of documentation in each case is generally superb.

Catching a Bus !

Neither chip is by definition tied to a specific physical bus architecture, but in each case a de facto bus organization has emerged. Note that not all 68XX computers use the S-50 bus nor do all 8080/Z80 computers use the S100 bus. Many major manufacturers have elected to not incorporate a bus design into their computer as a cost and space saving decision. For example neither the Radio Shack Z80 computers or the Xerox computers are bus oriented. If the physical sense of the word. Such computers almost invariably suffer from a lack of flexibility and expandibility as a result. I can not debate the merits of S-50 vs. S100 from an architectural point of view, but it must be noted that the S100 bus has been accorded by the IEEE as an industry standard. And lastly, it must be realized that therere of horrors!!) there is no reason why a 68XX chip can not be placed on the S100 bus. In fact, I know of at least one manufacturer (Ackerman Digital) which has done just that.

Anyone for Assembler?

If you are anticipating doing much assembler work, I assure you that the 68XX world is preferable from a development point of view. The problem with the 8080/Z80 assemblers goes something like this. First Intel (who manufactures the 8080 and 8085) defined the mnemonics and assembler syntax for the 8080 in an arbitrary fashion as of course was their right to do. The only problem is that they did a horrible job of it. Now Zilog (who is the manufacturer of the Z80) introduced their superset 8080 and called it the Z80. Clearly for the instructions that did not exist on the 8080, Zilog needed to introduce new mnemonics. However, they rightly decided that the Intel mnemonics were less than ideal and therefor came out with an entire new and much better list of mnemonics which were totally incompatible with the original Intel list. Of course the software houses as well as both Intel and Zilog wrote assemblers to support the various mnemonics. At least one such software house made further changes and introduced more confusion. Gary Kildall who is president of Digital Research and an ex-Intel man has doggedly stuck with the Intel mnemonics and indeed all the assemblers from Digital Research steadfastly refuse to assemble the Zilog mnemonics unless you get a macro definition library that defines the Zilog mnemonics as macros and expands them into appropriate Z80 instructions. Of course you need the macro assembler to do this, and of course the assembler provided when you buy CP/M is not the necessary macro assembler and on and on it goes. Needless to say, I have 4 or 5 assemblers that I use when working with CP/M. Some support Intel, some Zilog, one supports both, some support macros, and two produce relocatable linkable code. Believe me, there are days when I long for the straightforward TSC assembler.

How Fast Will It Go ?

Having said so much good stuff about the 6809 elegance and architecture, you might then assume it outperforms the Z80 in processor speed benchmarks. Well, not always. For purposes of comparison, I will reference the informal benchmarks conducted over a period of two years by Jim and Gary Gilbreath the results of which were originally published in the September 1981 (P.180) issue of Byte magazine, and recently updated in the January 1983 (P.283) issue of the same magazine. Where possible I have personally confirmed the results on my own hardware.

Gilbreaths' benchmark used a variation of the classic Sieve of Eratosthenes algorithm for computing all the prime numbers between 3 and 16,381. Other than the fact that the algorithm was chosen to avoid division, the algorithm itself is arbitrary as this was not a contest to determine the quickest way to compute n prime numbers. The algorithm was coded as consistently as possible in each of several languages and compiled and / or interpreted as necessary on many different processors and computer systems. In order to get a more meaningful result, the algorithm was repeated 10 times, however for the extremely slow machines or languages, the algorithm was run only once and the resulting times multiplied by 10 to allow proper comparison. I will constrain myself to summarizing 68XX and Z80 results although everything from Apples to the Cray-1 super computer were exercised. Note that other than assembly language, the results are a reflection upon the relative efficiency of the compiler or interpreter as well as providing a comparison of the processors.

	6809 2Mhz	Z80 4Mhz	6Mhz
assembler	5.10 sec	6.80 /	4.50 sec
IMS Pascal (compiler)	8.70 sec		
Introl C (compiler)	11.00 sec		
TSC Pascal (Uniflex compiler)	34.00 sec		
TSC Pascal (Flex compiler)	54.00 sec		
IMS Pascal (P-code)	105.00 sec		
BASIC 09 (OS-9 interpreter)	238.00 sec		
Dynasoft Pascal (P-code I think)	306.00 sec		
Lucidata Pascal (P-code)	735.00 sec		
TSC XBASIC (Flex interpreter)	840.00 sec		
FORTRAN (Microsoft compiler)		13.90 /	9.20 sec
PL/I (Digital Research compiler)		14.00 /	9.33 sec
CB80 (Digital Research Basic Compiler)		15.70 /	10.46 sec
Pascal MT+ (Digital Research compiler)		19.00 /	12.66 sec
UCSD Pascal (P-code)		239.00 /	159.33 sec
JRT Pascal (P-code)		383.00 /	255.33 sec
CBASIC (Digital Research interpreter)		484.00 /	322.66 sec
CBASIC (as above but all reals)		1430.00 /	953.32 sec
COBOL (Microsoft)		5115.00 /	3410.00 sec

The above chart lists popular 6809 and popular Z80 language systems with their respective results. Now, I suggest that no one take all of the above too seriously as there is a certain amount of apples and oranges comparison here. And of course, one small algorithm hardly comprises a thorough evaluation. However, some interesting observations can be made. First of all, Cobol does not belong on 8 bit computers unless you have a lot of time on your hands.

At the assembly level, the 6809 is very competitive with the 4Mhz Z80, in fact substantially quicker on a percentage basis. But when we compare the compilers and interpreters, it becomes apparent that the Z80 users are most often enjoying a huge speed advantage. Nearly all of the well known and widely used Z80 compilers produced results in the 14 to 20 second range when run on the 4 Mhz Z80. Yet on the 6809, only the Introl C compiler and the IMS Pascal compiler were in this league and in fact again somewhat quicker. The TSC Pascal compiler while not comparable to the Pascal MT+ compiler was at least reasonable. The real price/performance winner in the Pascal world is the JRT P-code implementation which comes as a complete package for $29.95. In defense of Lucidata, I believe (at least I hope) that the run was made on a 1 Mhz 6809 as this was not made clear.

What conclusions if any can be drawn? It appears that the 6809 is quicker than the 4Mhz Z80, but not quite as quick as the 6Mhz model. How about a 4Mhz 6809 Mr. Motorola? Also, and I must word this carefully, it appears that the Z80 CP/M user has a broader range of very well implemented language systems from which to choose. Now I know that some of the 6809 compilers produce position independent code and some of it is re-entrant and all of that may impact performance, but most of the CP/M languages allow (or demand) a linkage step that will relocate the code for you if you so desire. The 6809 user has most of the popular languages from which to choose, but amongst those, there are fewer really excellent implementations. The implementation is very important - just compare the IMS P-code Pascal results with other P-code implementations on the same processor. And lest the compiler builders declare war on me, I should also say that the speed of the resulting object code is only one measure of the quality of the compiler. Speed of compilation, size of resulting executable code etc. are other important considerations when comparing compilers.

The Operating Systems

Unfortunately, my comments must be directed toward Flex on the 68XX side and CP/M release 2.2 on the Z80 side as my experience is limited to these two. But first a few words about the other most often used operating systems on these machines.

While I have not used either OS9 or Uniflex, my understanding from discussing them with those who have indicates that they are way beyond either CP/M or Flex in many areas largely as a result of successfully implementing much of the Unix environment. Also in the "way beyond" category is an 8080/Z80 operating system known as OASIS. All of these more advanced operating systems allow multiple users and tasks and also implement a more sophisticated file system. Actually, CP/M itself has a multi-user big brother known as MP/M. Not nearly as widely used as CP/M and requiring multiple banks of memory to accommodate the multiple users, MP/M offers little in the way of features beyond CP/M. There have been a few CP/M clones over the years, the first I believe being CDOS from Cromemco, and more recently a version known as TurboDos which features much improved disk I/O at least from a speed point of view. And of course, Lifeboat Associates distributes CP/M under its own name SB-80. That leaves only the much maligned TRS-DOS (more often referred to as trash dos) which Radio Shack supplies with its Z80 computers.

Back to Flex and CP/M. As a blanket statement, it has been my experience that Flex is easier to implement, use, and train non-computer people to use. Indeed, CP/M at times is seen to be so convoluted that at least two companies market major products designed to insulate the casual user from the vagaries of CP/M. These products (Supervyz and Organizer) achieve this by presenting a series of menu displays which are customizable. They allow the user to select by number the function they would like to use which is then translated into CP/M computerese.

Both o/s provide a single directory disk, but only Flex has the date of last update maintained in the directory. Both allow attaching attributes such as suppress directory listing to individual files. But only CP/M allows a whole diskette to be write protected under software control as well as (optionally) hardware control. A major failing of CP/M is a provision to search for filenames similar to Flex's concept of system and work drives. Flex itself is maintained as a file under the file system which is read at boot time, whereas CP/M is stored physically upon the first 2 tracks of the diskette. There are two problems with this: first the code comprising CP/M can not be treated as a file by the system utilities thereby adding the necessary special utilities to maintain this code and its special area of the disk, and second, this disk space is forever wasted even if CP/M itself is not on the diskette.

CP/M allows arguments to be passed into its "submit" files (similar to "exec" files) which was a major shortfall in Flex, but on the other hand does not offer a "startup" facility. The pain involved in making our CP/M to do an automatic startup would leave a Flex user aghast. Also missing is the equivalent of Flex's "ttyset" facility. No quick file builder like "build" either. Instead you get the infamous CP/M editor called Ed. The huge sales of Wordstar and other good editing utilities should give you an accurate impression of just how many people are willing to use Ed. (It reminded me faintly of the old Motorola editor way back in the early Mikbug days.)

I always liked the Flex spooling mechanism (after I got it working), but CP/M doesn't have one of those as standard running gear. There is one for an extra $50 or so, but having once seen it implemented, I've never bothered buying it as it is convoluted and inferior to Flex's spooler. While we are on the topic of output, try to imagine using Flex without either the "O" or "P" commands to re-direct output for you. Impossible you say? Not at all! Hundreds of thousands of CP/M users get along without it! To be fair, you can cause console I/O to be copied onto the printer, but that is no substitute for I/O re-direction. This is possibly CP/M's greatest shortcoming.

On the plus side, is CP/M's famous "PIP" utility which replaces list, copy, append commands and still has features to burn. Briefly it copies (append is treated as a special case of copy) data from any source to any destination. The source and destination may be any physical character oriented peripheral or a disk file. PIP allows you for example to type from the console keyboard to the printer, a modem, a diskfile, or to copy a diskfile to the console, or printer, or modem or to another disk file etc. PIP and several other utilities allow specification of incomplete file names for the purpose of name matching, but perhaps PIP's greatest feature is the filters it can use on the data during the data transfers. Such as removing formfeeds or the parity bit in ASCII data, character case translation, expanding tabs, providing pagination, adding line numbers, etc. Flex has nothing even close to PIP.

CP/M supplies a utility called "DDT" which is a debugger and somewhat similar to the monitors we all expect to find living in EPROMS within our computers. DDT is useful insofar as it works with the file system as opposed to the usual sector level I/O found in most EPROM monitors.

Due to the way in which the CP/M memory map is arranged, CP/M must be at the top (high addresses) of configured real memory. But unfortunately it can not relocate itself at boot time and thereby adjust to a system with less memory. Therefor one also finds with CP/M a few utilities involved in trucking CP/M around in memory for other memory sizes. Having built a memory image of your new CP/M, another utility saves it out onto those two infernal tracks mentioned above for posterity. While on the subject of memory maps, I must say that although I prefer the Motorola style of memory mapped I/O (ACIA or PIA devices etc.), from a programmers point of view, the Intel style doesn't cause the memory map to be hacked up simply to allow for the presence of addresses which are to be decoded by I/O devices. With the Intel approach we get a clean contiguous 64K which CP/M is quite happy to use. A number of CP/M systems also use a "shootable" EPROM which after booting, turns itself off thereby allowing the full 64K of memory to be addressed. No reason why this could not be done on the S-50 systems either.

File management is very similar from a user's perspective on both systems. Both provide sequential and random file organizations with sequential and relative record access methods. One very major difference is the way Flex and CP/M implement the file system. Flex defines a file as a linked list of sectors while CP/M allocates file space with a bit map technique. Which is better? Well I will say that I think Flex reads and perhaps writes sequential files quicker, but CP/M may have the edge with random access. In any case, the speed differential was not that great based upon subjective observations not empirical measurement. However one thing I can say is that Flex diskettes are much more susceptible to corruption due to broken chains in the sector linkage. I have NEVER had a CP/M disk damaged. Both systems use the same disk drives (Shugart 801 series) so that is not a factor, however I admit the controllers are different.

At the system service level Flex's DOS routines and CP/M's BDOS routines while implemented quite differently, largely provide the user with the same type of functions and are similar when it comes to ease of use. TSC has however extended DOS to include all sorts of wonderful little goodies such as routines to parse the input buffer. CP/M is much weaker in these areas.

From the Ground Up!

I wonder how many of you have used TSC's general purpose version of Flex to implement your own Flex system from the ground up? I have! and it was both fun and quite easy. Nothing fancy about TSC's documentation, but it sure gets the job done. Flex's interfaces to the outside world are contained in two modules: the console I/O driver package, and the disk driver package which you must supply. This interface is the picture of simplicity and is very well defined.

If you have put up Flex in that style and are looking for a new challenge, may I recommend doing the same thing with CP/M. This is a job for those with stout hearts and a penchant for documentation originally written in Swahili or whatever. (Noted to those of you with Swahili as your native tongue, I apologize for comparing it with Digital Research's documentation). Functionally, you need to do all the same things! test your code, integrate it, write a loader, cross your fingers, and if all went well..... It just seemed an awful lot harder with CP/M.

Utilities, Languages, and Stuff!

Both operating systems have available several other utilities and languages. TSC at a reasonable cost makes available a large set of programmers utilities and [illegible] utilities. On the CP/M [illegible] because of its far greater [illegible] [illegible] [illegible] and its

brother has another language, editor, modem/communications package, sort/merge package etc. for sale. However one of the greatest sources of fine utility software is the CP/M User's Group. Now I know that Ron Anderson and his columns (which I very much enjoy) are the core of the Flex user community, and that through the pages of 68 Micro Journal plenty of utility software is published (and even available on disk from 68 M.J.) but folks, the CP/M group is MASSIVE. To date I believe they have distributed something in the order of 80-90 diskettes of public domain software. As you may imagine 75% of this is of less than commercial quality and often poorly documented etc., but the occasional gem is wonderful. For example there are thirteen or so languages, numerous directory display programs, a complete spelling checker, dozens of games, business programs etc. If there is a Programmers Hall of Fame, then a guy by the name of Ward Christensen who seems to have written a great deal of the truly good stuff and unselfishly gave it to the User Group should be inducted without delay. Being able to access this wealth of goodies is quite important to me.

As I noted earlier, CP/M provides you with the basic Digital Research 8080 assembler, but its usefulness is somewhat minimal. As indicated in the discussion on the speed benchmark results, I believe that on average the CP/M user has a much wider selection of quality language and utility software from which to choose thus allowing you to select the one that has the features you wish at the price you can afford. I've often said that the only reason I endure CP/M at all is just so that I can gain access to all the superb CP/M software. There have been a few comments passed in recent months regarding Wordstar in Ron's column. Wordstar whether you like it or hate it set the industry standard for full screen word-processing editors by which all others now appear to be measured. In Ron's case, he compares it to Stylograph. Ron's comments while correct are also misleading to a degree. It for example is possible to reformat the entire document with one command, but I admit there is a heck of a lot of control sequences to learn. In any case, nobody says Wordstar is the "best"! something which is "first" usually never remains "best" as others come along and improve upon the original concepts. There are now several WP packages for CP/M which claim to have more features and ease of use than Wordstar. My only comment is if you don't like Wordstar or Stylograph or whatever, then find one you do like AND USE IT! Which one isn't important! the productivity gains they provide are. (This is being prepared with Wordstar).

I love the languages I can buy for CP/M. If you like Basic and are not embarrassed to use it after all the hate articles it has inspired over the last two years, CBASIC and its big brother CB80 which is a superset of CBASIC'S syntax plus fully compiled is the greatest thing since sliced bread. Microsoft who got to be famous by implementing Basic, can not touch this stuff! Sorry to say, neither can TSC's XBASIC. At the far edge of the other side of the ideological spectrum there are a couple of extremely good Pascal compilers, and even more good P-code ones. Personally, I am in love with Digital Research's PL/I compiler which combined with the linkage editor lets me do anything I can do on my mainframe at work, and often outstrips my minis. For those of other inclinations, there are dozens of Forths, a few ADAs, APL, COBOL for those foolish enough to try and use it, Fortrans, LISPs, close to a dozen C compilers (my favorite for system software), literally scores of Basics, etc.

It seems that between Flex, Uniflex, and OS9, the 6809 user has access to most of the popular software items of the day ie. screen editors, spelling checkers, languages, sorts, spread sheet calculators and so on, but at the risk of being repetitive, I must admit CP/M users have a far greater choice and perhaps on occasion, better implementations.

The Future

Where do each of our two environments go in the future? At the moment, Digital Research is releasing CP/M release 3.0 which by all accounts fixes and enhances CP/M to the point that it will compare at least favorable with Flex. Specifically, many of the complaints I have registered above have been addressed. However, I think it is safe to guess that release 3.0 may be the last release of CP/M for the Z80 as the 16 bit machines are attracting most of the attention of the software developers - Digital Research included. CP/M has now been implemented for both the Intel 8086 and the Motorola 68000 families. The subtle notion that somehow this is upward compatible from CP/M on the Z80 is totally false. Only the user interface and disk formats stay the same although a program (non-assembler of course) would be transportable at the source level. CP/M will not likely ever dominate the 8086 world in the way it completely ruled the 8 bit 8080 machines. Already, Microsoft's MS-DOS (PC-DOS if you prefer) is leading the way in this arena. MP/M is already dead, and I can't really see anyone wanting to run CP/M on a 68000, why bother when far superior operating systems including full Unix are available. Presumably Uniflex will as per TSC's promise finally run on the 68000 as well.

In a similar vein, most of us would agree that Flex as we know it has reached its maturity and should not grow much farther. Instead, we have as alternatives OS9 and Uniflex which are very solid operating system environments. Functionally, these two are the class of the 8 bit world, and are an excellent transition tool into the 16 bit future. Hopefully our two primary software vendors TSC and Microware will build upon these operating systems and eventually build a bridge to the 68000. For those of us who want the extra horsepower and functionality of the 16 bit processor, there is no question at all where the real future is to be found - the Motorola 68000. Already, the choice of machines based upon this chip and the operating systems which are available to support it are just simply incredible.

incredible selection of software and the access to the CP/M User's Group that I get with CP/M. Since CP/M is so widely used, the CP/M disk format recorded in single density on 8" drives has become another de facto industry standard for software distribution.

But if you can have only one of the two! well, I guess you will just have to decide for yourself!

I've only touched on highlights! if you have further insight or opinions I'd be most interested to hear them either in the pages of 68 Micro Journal, or addressed to my above address. If sufficient interest is indicated, I am more than willing to narrow in on specific areas of comparison in a future letter (or is this an article) should Don Williams be interested in publishing this information for his readers.

Summary

So who wins? Which is the best? The answer, of course as it always is in this kind of complex comparison is that neither is universally better or worse. It depends on what is important to you and the use for which you intend to use the computer. It is almost always true that an end user who sees the computer only from the perspective he or she can gain from sitting at a terminal entering the day's G/L postings can not tell what kind of computer is driving the application program. The business man hardly cares if it is a Z80 or 6809, whether or not the bus is S-50 or S100, whether the application is written in Pascal or C. Does the machine successfully achieve the use for which it is intended in a cost effective fashion seems to be the important question.

From a hacker's point of view, I prefer the elegance of the 6809, the ease of use of Flex, TSC's documentation, and the promise that OS9 and Uniflex hold for the future. I also like the selection of hardware I can buy because I have a S100 bus computer, CMOS [illegible], music boards, colour graphics boards, Z80, 8080, 8085, 8086, 8088, 6809, 68000, 16032 processors! all on that one bus. I also like the

MODIFYING OKIDATA 84 PRINTER

to Accept DECWRITER LA-36 Ribbons

by Bud Pass

Introduction

The OKIDATA 84 printer is fast (200 CPS), reliable, flexible, etc. It has one irritating problem in that it uses a reel-to-reel ribbon mechanism, which is not as convenient as cartridge ribbons, and uses ribbon reels which are currently hard-to-get and expensive.

Since the Digital Equipment LA-36 printer is so extremely popular, the ribbons for it are much cheaper and easier to get than the ribbons for the OKIDATA 84. Unfortunately, the ribbon reels are not compatible, although the ribbons themselves are. One solution is to unwind the used ribbon from the OKI reel and to wind new LA-36 ribbon onto the freed spools. Obviously, this is messy and time consuming to perform without the proper tools. A better solution would be to modify the OKIDATA to accept the DEC reels directly.

The following discussion outlines the mechanical conversion of an OKIDATA 84 to use DECWRITER LA-36 ribbon reels. Please note that this conversion will not work with the OKIDATA 82 or 83 printers, since they use smaller ribbon reels. It also voids any warranties. With the right tools and parts, it should take only about one hour.

Conversion

You will need the following parts, along with a standard tool kit:

- -two cylindrical brass or hard plastic shims 1" long, 1/8" ID, 3/8" OD,
- -enough firm material 1/8" thick for two 3" diameter circles (may be cardboard, PC board, etc.),
- -four small-headed 1" 4/40 bolts and washers and eight matching nuts.

Ensure that you have these parts before starting the conversion, or your printer may be down longer than you expected. Remove power, the loose plastic cover, and the ribbon from the printer. Move the head to the extreme right or left margin. If you feel it helpful, remove the top cover of the printer.

Place the shims over the ribbon reel shafts and LA-36 reels over the shims. They will not fit well because of the keys on the nylon gears, but hold the reels horizontal and mark the gear through the key holes on the reels. A mechanical pencil with lead over-extended may be used for this task. Remove the shims and reels.

Carefully remove the retaining rings from the shafts which support the ribbon reels, and work the washers, springs, and gears free from the shafts. All parts are symmetrically placed on the two shafts, so there is no right-left ambiguity of parts replacement. However, be sure not to lose any parts, as they may be difficult to replace or may cause mechanical failure if they land between two opposing moving parts.

Drill out the expanded hollow metal shafts holding the reel keys in the nylon gears. Try not to get the gear too hot, as it will melt. Pull the keys from the front. Using a bit the same size as the 4/40 bolts, drill the new key locations at the sites marked previously. Using a much larger drill, form countersink depressions in the back of the gears just deep enough so that the heads of the bolts will not protrude. Insert the bolts into the gears from the back, screw them completely into the gears, and put nuts on the top to secure them firmly to the gears. Ensure that the reels fit easily onto the bolts, or make adjustments in the bolt positioning until they do fit.

Cut out two rings from the 1/8" thick material, with approximately 1 1/2" ID and 3" OD dimensions. Slip it over the fronts of the gears and trim if necessary to ensure that it lies flat on the gears and does not extend beyond the inside of the teeth on the outside of the gears. Attach with thin tape or glue to the gears.

Slip the gears back on the shafts, the shims back on the shafts, and the reels back on the shims. Ensure that the reels fit easily and do not strike the ribbon guides, the ribbon driver between the reels, and the head. If one strikes a ribbon guide, file the offending part of the guide until the reel does not strike it. If one strikes a ribbon driver, pad the 1/8" thick material evenly with vinyl tape until it clears. If one strikes the head (with the adjust lever in the number 2 position) bend the vertical ribbon mechanism evenly on both sides until both reels clear, keeping the top of the ribbon mechanism level.

Remove the shims and reels and replace the springs, washers, and retaining clips on the shafts. In case you did not observe the original order, the springs go first, followed by the large metal washers, the small mylar washers, and the clips. Replace the shims back on the shafts.

Re-install the ribbon on the LA-36 reels through all the guides used before except for running the ribbon on the opposite side of the ribbon guides adjacent to the reels. This is necessary because the LA-36 ribbon winds onto the reels at a large enough radius that the ribbon would otherwise miss the adjacent ribbon guides when the reel is nearly full. Install the 4/40 nuts on the bolts to secure the reels; if necessary, use the washers to prevent the bolts from slipping through the holes in the reels.

If the ribbon does not pull straight through the adjacent guides, pull up on the offending guide with a pair of pliers while rotating it slightly. If this does not correct it sufficiently,

file a little of the metal off the guide until it does fit. If you have done any filing here or earlier, be sure to collect all scrap metal filings.

At this point, the ribbon should run fairly easily from one reel to the other through all the guides, but not spin easily enough to leave slack at any point. If it does not, correct the problem.

When you are ready, put paper in the printer and run a self-test by holding the LINE FEED button while turning the power switch on. Be ready to turn power off quickly if the head snags the ribbon or you hear loud noises from the ribbon driver gears or from the head striking the reels. The reels should ride flat and smooth on the gears. Because the ribbon winds at a larger diameter on the new reels than on the old ones, the speed of the ribbon through the guides will be increased, and this will increase the strain on the ribbon and guides somewhat. If necessary, bend the guides at the extreme margins slightly to ensure that the ribbon feeds smoothly and evenly without bunching.

Assuming all is OK, turn the power off, and remove the nuts securing the reels to the gears. Pull up on each reel. If the center shim comes up with the reel, remove the reel and shim. Wrap enough vinyl tape around the shaft (start with one turn) so that the shim does not easily turn on or come off the shaft; do not use glue, as that would prevent the removal of the nylon gears. Replace the reels on the shims and resecure them with the nuts (and washers, if used). Note the revised ribbon path on the inside of the loose plastic cover and replace all covers previously removed.

Summary

The preceding has described how to modify the OKIDATA 84 to use DEC LA-36 ribbons and reels. Such a modification has the advantage of making the replacement of ribbons simpler and more economical.

20 Box 6
Gray Court, SC 29645

April 13, 1983

The Editor
'68 Micro Journal
5900 Cassandra Smith Road
Hixson, TN 37343

Sir:

We are pleased to announce the Metropolitan Greenville (SC) Color Computer Club formed in January of this year and already almost fifty members strong.

The club serves the interests of present and prospective CoCo owners in the entire western South Carolina region. As a group, we are totally committed to computer literacy among ourselves and within the community. Members enjoy a lively exchange of computing information, free software, programming, and hardware tutorials as well as a bi-weekly club newsletter.

Meetings are held every Thursday night at 7:30 at the Plain Elementary School in Simpsonville, SC.

Anyone wanting more information about this dynamic organization may contact us any time at (803) 676-3328 or 3812, or write.

ED LOWE
Sec'y/Treas

ACORN™ COMPUTER SYSTEMS 68-SOC

April 14, 1983

Dear Don:

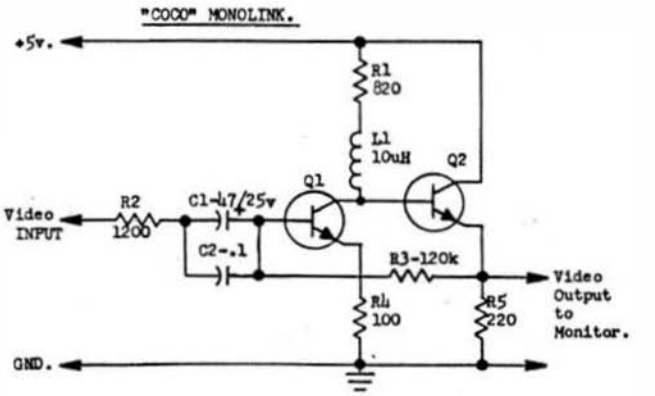
We have wanted, for some time now, to use our "COCO" with a high quality monochrome monitor. We tried many of the published circuits, but the results were always disappointing. So, we designed our own.

Our circuit has been tested on an 18MHz monitor in both the normal character and bi-rez graphics modes. The quality, sharpness, and definition of display must be seen to be believed. Note that this circuit does not disturb the RF modulator system of the "COCO". In addition, both your color television and a monochrome monitor may be used at the same time.

If we receive enough requests, we will offer the "MONOLINK"(TM) in kit form. You may publish this letter and information for the private use of "COCO" owners. We reserve the rights associated with this design for manufacture.

Thank you,

Ross Blank
ACS, Inc.



All resistors ¼watt.
Q1-TC0108 Q2-TC0128 L1-J.W. Miller#70W10SA1

INPUTS to MONOLINK from "COCO" video mixer MC1372 chip.
+5 pin 11
Video pin 9 > MC1372
GND pin 4

ACS INC

ACORN™

COMPUTER SYSTEMS SS-50C

SIMPLIFIED PIA CONTROL REGISTER INFORMATION
FOR USE WITH STANDARD HANDSHAKING

CTL LINE	CR BIT	0=CLEAR 1=SET
CA1 CB1	0	0=IRQ DISABLED, ALWAYS HIGH 1=IRQ ENABLED, GOES LOW ON SIGNAL
	1	0=HIGH TO LOW TRANSITION, ACTIVE 1=LOW TO HIGH TRANSITION, ACTIVE
DDR	2	0=ACCESS TO DATA DIRECTION REGISTER $00=INPUT, $FF=OUTPUT 1=ACCESS TO I/O PORT BUFFER
CA2 CB2	3	WRITE AN 0 TO SET CX2 LOW WRITE AN 1 TO SET CX2 HIGH
	4	1=STANDARD HANDSHAKING
	5	1=CX2 SET AS OUTPUT LINE
CA1 CB1	6	NOT USED, READ ONLY
	7	IRQ FLAG (READ ONLY)

ACORN COMPUTER SYSTEMS

April 23, 1983
7204 E. 28th St.
K.C., Mo. 64129

'68' Micro Journal
5900 Cassandra Smith Rd.
Hixson, Tn. 37343

Dear Don Williams

Enclosed you'll find a check for a subscription to '68' Micro Journal. Just need to say a couple of things.

1) does anyone know what ever happened to the MC6800 User's Group from Motorola. I tried to apply and my program came back to me.

2) I'd like to make a special comment to one of your advertisers. Midwest Scientific Instruments of Olathe, Ks. I had just ordered a couple of boards from the company and was really surprised at the price of the boards. When the bds came in I was in shock to see the quality of the boards. I have never seen a better built board. Needless to say MSI will always have my business. The gentlman who answered the phone not only sold me the boards but he also was alot of help.

One more thing to comment on, is that it seems that more and more manufactures are finally coming around to the Motorola CPU's. I'm glad of that fact. So instead having to write everything ourselves, we can now maybe buy some packages.

Best regards,
Adrianus C. Hoogenraad

JIM SCHREIER
4327 East Grove Street
Phoenix, Arizona 85040

Dear Don,

Attempting to convert a Program to TSC's Basic recently proved to be a real challenge. Walking lightly may be the key to using some of TSC Basics' features, but this patch of quicksand came as a surprise. It seems to be in all editions and version of TSC's Basics, including the 6809 release 22 of 12/04/81.

TSC's Basics divide in an inconsistant manner. Basic fails to perform internal tests allowing the same displayed number to be unequal. This happens for the number 1 as well as some other numbers. It seems that the number one is stored differently in memory, one number may be stored positive, the other, negative. Nevertheless, they are displayed as equal, but test unequal.

Louis Boyd isolated the problem. These two routines illustrate the difficulity and one possible solution.

```
 5 REM Routine A
 6 REM Incorrect  Division Demo
10 DIGITS 17
20 FOR N=1 TO 20
30 M=1*X/X
40 M=1/X*X
50  PRINT N,M,
60 IF M<>N THEN PRINT "NOT EQUAL";
70 PRINT
80 NEXT X
```

```
 5 REM Routine B
 6 REM Solution to Incorrect Division
10 DIGITS 17
20 FOR N=1 TO 20
30 M=1*X/X
40 M=1/X*X
41 M=M-1+1
50 PRINT N,M,
60 IF M<>N THEN PRINT "NOT EQUAL";
70 PRINT
80 NEXT X
```

Since this appears to be an actual "bug", other TSC Basic users may wish to contact me at (602) 276-6216 or Louis Boyd, 629 North 30th Street, Phoenix, Arizona 85008; (602) 275-8258; with their comments and observations.

Best regards,
Jim Schreier

John P. Tucker
12015 O'Connor Road #184
San Antonio, Texas 78233

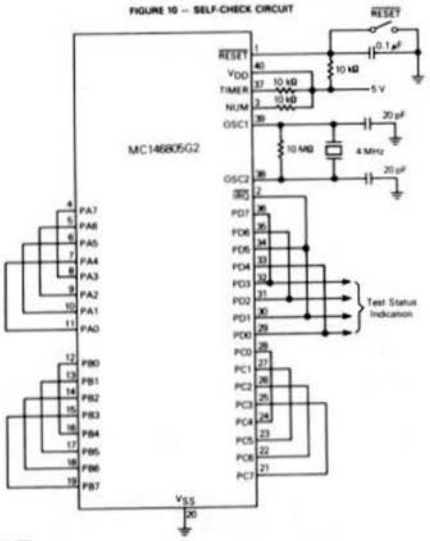
REVERSE, a game from Micro-Power; 1418 Thorndale; Chicago, Illinois 60660. "(Giggle!) I beat the computer -- again." You'll be hearing that often when you play this delightful little Othello-like game from Micro-Power. And you'll play it often because of the good graphics it uses. My version was for a SWTPCo 8212 terminal that I always run at 32,400 baud. Since Reverse runs in TSC's Extended Basic, the screen changes as fast as information can come out of the black box on my left. REVERSE follows all of the rules of Othello, but the mathematics used are set for it to play a relatively low level of difficulty game. (I'm sending Micro-Power my own version in hopes they'll incorporate their clever presentation into a stronger game.) However, each move is displayed in large, easy-to-see "playing pieces" and the actual changes and additions are displayed instead of the screen being cleared and re-built each time. This is a highly desirable feature and the authors are to be congratulated on the manner in which this is accomplished. Also enjoyable is the fact that scoring is updated with each move. The program is distributed in compiled form only, thus it was not really practical to dig into it to make changes. Perhaps Micro-Power would like to review this policy on their games programs. I enjoyed the game and commend it to your attention, keeping in mind that it is more fun to play against a human opponent than it is against the computer, in its present form. The program allows you that choice. (The rules of Othello are so widely known that they are not mentioned here to conserve space. The documentation is simple and occupies part of one side of one sheet of paper, this including the explanation of the features such as SKIP to skip a turn or TAKEBACK to takeback a turn that shouldn't have been made, etc. Yet, the documentation is adequate.)

ERRATA
to
MC146805G2
ADVANCE INFORMATION DATA SHEET
(ADI-884-R1)

In the wiring of the external connections needed to perform the self-check procedure, the interrupt pin (pin 2) should be connected to PD5 (pin 34) and not PD6 (pin 35). This correction was erroneously shown in the self-check circuit diagram of the MC146805G2 Advance Information Data Sheet. The following diagram has been corrected and replaces Figure 10 (page 9) of the data sheet.

The change affects only those using edge- and level-triggered interrupt pins. The change will not affect the operation of the user-definable subroutines as they require no external connections.

FIGURE 10 — SELF-CHECK CIRCUIT



ADI-884-R1A1

©MOTOROLA INC., 1982

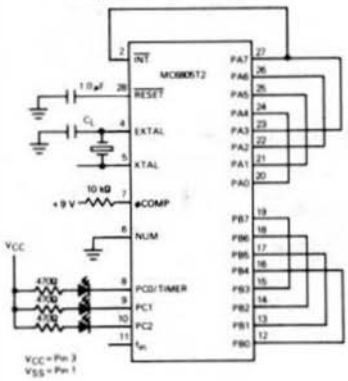
MOTOROLA Semiconductor Products Inc.

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.

ERRATA
to
MC68B572
ADVANCE INFORMATION DATA SHEET
(ADI-887)

A 10 kΩ resistor in series with 9 volts is required from pin 7 (comp) and not, otherwise catastrophic breakdown occurs on pin 7. This resistor was omitted from the Self-Check Connections diagram found in the MC68B572 Advance Information Data Sheet. The following diagram has been corrected and replaces Figure 10 (page 9) of the data sheet.

FIGURE 10 — SELF-CHECK CONNECTIONS



ADI-887-A1

©MOTOROLA INC., 1982

MOTOROLA Semiconductor Products Inc.

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.

HOW TO BECOME A FORMER GIMIX DEALER

- **Don't Give Support.**
- **Don't Answer Phone Inquiries During Normal Business Hours.**
- **Indulge in "Creative" Software Packaging.**
- **Be Deceitful.**
- **Don't Pay Your Bills.**
- **Don't Keep Any Inventory.**
- **Engage in Bait and Switch Tactics.**

We At Gimix Will Not Tolerate Any of the Above. We Have Many Decent Dealers Worldwide Who Do Support and Serve Their Customers.

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510
TWX 910-221-4055

GIMIX inc.

HELP

HELP!! Since Percom has decided to become un-userfriendly to the 68XX after I had purchased Percom's INDEX (6800) but before I had any software for it, does anyone have software or the Advanced Users Manual for INDEX with which they would be willing to part?
Bud Modery, 2357 Hood Ct., Santa Rosa, CA 95401, (707) 575-7140.

***

CLASSIFIED ADVERTISING

SWTP 6800 W/MP-B2, 2-MP-C, 3 MP-S, 2 MP-LA, MP-T, MP-R, Microtime Bd., AC-30 Cass.Intf., F&D MDI-1 Disk Controller, much software. $1150 takes all or best offer.
Dave Balko, 3005 Concord, Trenton, MI 48183 (313) 676-6856.

***

Beehive 105 Terminal 24X80 19.2k VGC $250 U Ship
Dave Balko, 3005 Concord, Trenton, MI 48183 (313) 676-6856.

***

SWTPC 6800, 48k memory board, 8k memory board, 4k memory board, AC-30 cassette interface, LFD 400, Mini- os + X. 3 Serial Interfaces, Heath Kit printer, Newtech Musicboard, Modem plus much more. Best offer over $800
Mihran Kochyan (313) 271-3594.

***

Gimix Mainframe, 6809 CPU "Plus", DMA 5/8 Controller, 56k, 2-FDD 100-5 Drives, 8-port serial, 8-port parr, 80X24 Video, Voice Synt., Eprom-Pcb, Thomas-Modem, Thomas-Video, TT-Pcb, 2-FDD 100-8 In Vista-Cabinet V-1000, S-9, Flex-9 + Much more software.
2-Sanyo Dm 5112 cx Monitors, Teletype 43-Rs232 $5000 or best.
201-662-1826.

***

SWTPC Dc-3 DS/SD disk controller $100. Three DRC 16k SS-50 static ram cards $100 /best offer.
Tom J. Harmon, 15418 Diana Lane, Houston, Tx 77062 (713) 480-6075.

***

For Sale: SWTPC 6800/Ct-64 Terminal, Green Phosphorus Monitor and software $600/negotiable.
J. Sanders, Denver, Co (303) 733-9223.

***

200 8" Verbatim FD 34-9000 SS/DD disks. These disks have been formatted and written to one time. Make me an offer.
LEW (615) 842-4607.

***

SWTPC 6800, 48k, Soroc 120, 1-MSI 8" Disk Drive, SWTBUG, 8Ch A/D, 2 MP-LA, 2 MP-S, 2 DOS, MSI and Flex. MSI BAS 3.0, XBASIC, TSC ASSEM, Editor, Word Processor. Software Including Mailing List, Real Estate, Statistical, Financial, Mathematical. XTRA 4k BD + MPC, All manuals and documentation included. Fully operational system $2K.
Okidata M82 with system-$500.
Herzmark 602-867-3699, 2734 E. Cheryl, Phoenix, Az 85028.

***

Custom built 6809 Computer system includes SWTPC MP-MB, MP09A, MP-ID, DC-4C, with 56K, 2 Serial Ports, 2 Parallel Ports plus Printer Port. Software Includes Flex, Utilities, TSC Editor, Assembler, Debug, XBasic, Sort/Merge, Pascal, Fortran, Stylograph, Dataman and more. $1500 for all.
K. Kanakos, 17 Birch St., Lake rove, Ny 11755 (516) 588-2216 eves.

** QUALITY SOFTWARE NEEDED **

Standard S50 Bus and Color Computer

For the past few months we at the South East Media Division of Computer Publishing, Inc. (CPI), the parent company of 68 MICRO JOURNAL, have debated expanding into the software distribution business. Many other magazines have been doing so for years. Presently there are many fine examples of software that has been developed by **YOU** our readers, that will never see the 'light of day' unless someone, with enough exposure and willingness to continually advertise, runs with the ball.

Software is the 'backbone' for the **real** utilization of any computer, ours are no exceptions! Realizing that there will be some conflicts, with other advertisiers, this has been no simple decision. However, since day one the foremost concern of 68 MICRO JOURNAL has been it's **readers!** Therefore, South East Media Division will accept, for appraisal, software that runs on 6809 systems, games, utility or applications programs.

In the past there has been too much software offered that was not quite ready, nearly, but not quite. We will strive to eliminate that element. But right up front we tell you only that we will do our very best, nothing more. Also we will strive to keep cost to a bare minimum, while securing for the author a fair return, in royalty payments, promptly paid.

Of course we will expect, no — demand, that the author keep the product free of errors (bugs), and maintain it on a prompt and business like basis. Also we shall require that authors be willing to furnish 'source' for those programs that justify, by price and utility, inclusion of same. The lack of source code, properly commented, is a continual complaint we hear. Not all programs will be sold with source, but where necessary, we will insist that it be included.

In some instances the program may be small or short and not justify itself as a 'single' sale product. In this event it will be combined with other like programs, and offered as a package. In that event the royalties will be split between the various authors.

If you have software that you feel will qualify under this program please contact the proper person as shown below.

Color Computer	Standard S50 Bus
Tom Williams	Don Williams
Bob Nay	Bob Nay

Remember, if your software has **any** problems or 'funnies' — GET IT STRAIGHT BEFORE YOU CONTACT US!!!!!
Also get your source code in proper shape and well commented. There is too much 99% code already drifting around.

South East Media, POB 794, Hixson, TN 37343 — (615) 842-4601
A Division of CPI

FLEX™ USER NOTES
THE 6800-6809 BOOK

By: Ronald W. Anderson
As published in 68 MICRO JOURNAL™

The publishers of 68 MICRO JOURNAL are proud to announce the publication of Ron Anderson's **FLEX USER NOTES,** in book form. This popular monthly column has been a regular feature in 68 MICRO JOURNAL SINCE 1979. It has earned the respect of thousands of 68 MICRO JOURNAL readers over the years. In fact, Ron's column has been described as the 'Bible' for 68XX users, by some of the world's leading microprocessor professionals. Now all his columns are being published, in whole, as the most needed and popular 68XX book available. Over the years Ron's column has been one of the most popular in 68 MICRO JOURNAL. And of course 68 MICRO JOURNAL is the most popular 68XX magazine published.

As a SPECIAL BONUS all the source listing in the book will be available on disk for the low price of: FLEX™ format only — 5" $12.95 — 8" $16.95 plus $2.50 shipping and handling, if ordered with the book. If ordered separately the price of the disks will be: 5" $17.95 — 8" $19.95 plus $2.50 shipping and handling.

Listed below are a few of the TEXT files included in the book and on diskette.

All TEXT files in the book are on the disks.

LOGO.C1	File load program to offset memory — ASM PIC
MEMOVE.C1	Memory move program — ASM PIC
DUMP.C1	Printer dump program — uses LOGO — ASM PIC
SUBTEST.C1	Simulation of 6800 code to 6809, show differences — ASM
TERMEM.C2	Modem input to disk (or other port input to disk) — ASM
M.C2	Output a file to modem (or another port) — ASM
PRINT.C3	Parallel (enhanced) printer driver — ASM
MODEM.C2	TTL output to CRT and modem (or other port) — ASM
SCIPKG.C1	Scientific math routines — PASCAL
U.C4	Mini-monitor, disk resident, many useful functions — ASM
PRINT.C4	Parallel printer driver, without PFLAG — ASM
SET.C5	Set printer modes — ASM
SETBAS1.C5	Set printer modes — A-BASIC
	(And many more)

**Over 30 TEXT files included in ASM (assembler) — PASCAL — PIC (position independent code) TSC BASIC-C, etc.

NOTE: .C1,.C2, etc. = Chapter 1, Chapter 2, etc.

This will be a limited run and we cannot guarantee that supplies will last long. Order now for early delivery. Delivery is expected to start middle of June '83.

Softcover — Large Format

Book only: **$7.95** + $2.50 S/H

With disk: 5" **$20.90** + $2.50 S/H

With disk: 8" **$22.90** + $2.50 S/H

See your local S50 dealer/bookstore or order direct from:

Computer Publishing Inc.
5900 Cassandra Smith Rd.
Hixson, TN 37343
(615) 842-4601

™FLEX is a trademark of Technical Systems Consultants

SCREEN EDITOR/WORD PROCESSOR FOR DOS6x, FLEX, OS-9

IF YOU ARE CONSIDERING UPGRADING YOUR EDITING AND PROCESSING CAPABILITIES, HERE ARE A FEW GOOD REASONS TO CONSIDER SCREDITOR...

REASON #1 - SCREEN EDITING

Screen editing (sometimes called cursor-based editing) is the most efficient way in which to enter your thoughts into the computer. Since you see the actual text you are entering on the screen as if you were looking at a printed page, organization becomes a matter of observation rather than guesswork.

REASON #2 - DYNAMIC SCREEN FORMATTING

With dynamic screen formatting, the screen becomes an image of what the completed job will look like. Within the capabilities of the display terminal itself, you are constantly looking at what you would have if your job were printed. Every change you make, every addition or deletion, is automatically incorporated in a fully-justified, ready-to-print form. You no longer need to guess what all those cryptic processing commands you may have used with other software will turn your finished job into!

REASON #3 - NO MACHINE LANGUAGE PATCHING

SCREDITOR III is unique in that you NEVER have to make machine language patches to integrate it into your system. If you can answer a series of questions about your printer and terminal's capabilities, installation is complete.

REASON #4 - NO EXTRA "HIDDEN" PROGRAM COSTS

SCREDITOR III is complete as purchased. You get screen editing, formatting, mailing list merging and much more, all in a single program, and it's up. Even though the competition can provide many of the features of SCREDITOR III, how much will you pay? And, more important, how many programs will you have to buy?

REASON #5 - OPERATING SYSTEM SUPPORT

To our knowledge, SCREDITOR III supports more processor and operating system combinations than ANY other single editing and word processing package available. OS-9 (level one and two); Flex and SSB DOS, 6800 and 6809 versions are available. Except for those commands which are peculiar to a particular operating system, SCREDITOR III is identical in operation in all versions.

REASON #6 1/2 - OPERATING SYSTEM SUPPORT COSTS

If you purchase another company's product for your FLEX-9 system and decide to run OS-9, how much will it cost you for the extra copy of their software (if it's available. See reason #5.)? With SCREDITOR III, tell us you are upgrading and you pay 80% of the price you paid for the first copy and you can have the same great capabilities with both operating systems!

REASON #6 - CAPABILITIES

Not only is SCREDITOR III the only COMPLETE editing and word processing program available, it's also the most powerful. See if the other combinations of programs available give you true multi-column editing. Or allow you to add and delete; align rows and columns of numeric data. Or allow you to change margins and tabs at any time during editing just like a typewriter. Or give you commands to re-justify and change the width of paragraphs and lines on-screen, while you watch. Or edit and re-execute commands. Or allow essentially unlimited mixed text and command entry replacement with a single keystroke (macros). Or let you save all of those macros to a data file to be re-loaded for use in later edit sessions. Or edit your macros. Or...

REASON #7 - DOCUMENTATION

SCREDITOR III has the most comprehensive documentation available for any 6800 editor/word processor. The Technical Reference Manual is currently about 360 pages long. The Training and User's Guide we are now preparing will be about another 200 pages. We're also working on our new Tutorial Cassette (two hours of instruction). We even provide a HELP command which gives you the syntax of all of SCREDITOR III's commands and at least one paragraph of description on each!

REASON #8 - PRICE

Considering its capabilities, SCREDITOR III is the least-expensive program of its type available today, Period.

REASON #9 - HAPPINESS

We want your business. Think how happy you will make us if you decide to buy SCREDITOR III!

The following information should answer most of your questions about SCREDITOR III. If you have additional questions, please call during our office hours, or, better still, order the manual (when you decide to buy SCREDITOR III, we'll apply your manual purchase against the software price).

PROCESSORS SUPPORTED: - 6800, 6809

OPERATING SYSTEMS: - FLEX, SSB DOS, OS-9 (Level 1 and 2)

DISPLAY REQUIREMENTS - Serial terminal with computer-addressable cursor. 4800 baud minimum data rate, 9600 or better preferred. Clear EOL and clear EOS not required but highly desirable. Insert and Delete line improve display speed. Suggested terminals include Televideo, Beath/Zenith H-19, SWTPCo. Virtually all addressable-cursor terminals can be used. ANSI terminals supported by the time you read this.

Memory-mapped displays require software to provide terminal look-alike capabilities. Sample software included with FLEX and SOSKs orders.

Minimum display size is 16x64. Maximum is 255x255. No software patching is required to adapt software to display size.

Unique configurations may be supported to calls to external user-supplied sub-routines.

KEYBOARD REQUIREMENTS - ASCII character set with control codes required. Function keys generating Multiple-character sequences also supported. Control code and function key assignments user-defined and changeable at any time without patching.

KEYBOARD INTERFACE - Fit parallel, ACIA serial, or external keyboard handlers. Interrupt-driven operation (queue-buffered type-ahead) is a standard user-selected option.

PRINTER HARDWARE - Current version supports standard printers. Specialty printer capabilities controllable by transmission of control or escape sequences imbedded in text. Underline, boldface, double width simulated on printers with backspace capability. Proportional printing available soon.

PRINTER INTERFACE - Program uses existing user-supplied software to transfer characters from SCREDITOR III to printer hardware by calls to user sub-routines.

COMMAND SET - Fourty-six commands (or more by the time you read this) with multiple options on many commands. Thirty-one single key commands for often-used operations such as cursor movements, inserts, deletes, paragraph formatting, etc...

OPERATION AND DISPLAY - Text entry by 'window-type' screen/cursor editing. Commands to single-key or line entry. Continuous display of current tabs/margins, edit modes, cursor row and column, current line number, horizontal display range.

REQUIRED MEMORY - (program memory) - Program size - approximately 20 kilobytes. Minimum memory for useful editing - 32 kilobytes.

DISK MEMORY - Configuration files (3 minimum) - 1 to 2 sectors each
HELP file - disk-resident, approximately 10 kilobytes

********** AT LAST! **********

AVAILABILITY - Please allow one to two weeks for delivery. Disks are built as orders are received to allow the most current possible revision of software to be shipped.

PRICE - SCREDITOR III manual, disk $150.00
manual only (price may be applied to software order) ..... $ 40.00

IMPORTANT INFO

We accept Mastercharge and VISA credit card orders, personal and business checks drawn on U.S. banks and in U.S. dollars, and COD's to the continental United States. Domestic orders shipped by United Parcel Service unless otherwise requested. Overseas orders shipped by air parcel, COD's shipped by United Parcel Service only. Open accounts to government agencies, educational institutions, established accounts and Fortune 1000's only. Virginia residents add 4% state sales tax. On orders under $500, add $5 for shipping and handling. On overseas orders add $10 for shipping.

THE BEST NEWS!

It is our haste to do business, too often the temptation to compromise our faith weighs heavily. Even though we have forgiveness when we confess our sins from a God who loves us, we would like to take a moment to ask forgiveness from any of our customers whom we have slighted. Orders do get lost or mis-shipped, and mistakes do happen. If we fail to completely please any of you in either our response to your order, inquiries, or for whatever reason, let us know. We will do anything in our power to satisfy your problems. We thank all of our customers for your prayers, and we pray for each of you that you will come to know the love of God in your lives that we have found!

ALFORD & ASSOCIATES
P.O. BOX 6683
RICHMOND, VA., 23230
804-320-6722

CoCo SPEECH SYNTHESIZER

The Color Computer is one of the best things to happen to computing since the 86-90 bus! At the price, there is no better way to get into personal computing, and we are now offering a speech synthesizer for the CoCo! With most of the same features of our VS-1, the SP-1 can give your CoCo the power of unlimited speech. Unlike software approaches to speech, the SP-1 does not eat your memory, and provides the clearest, easiest to understand speech we have heard!

The SP-1 is a completely assembled and tested unit which plugs directly into the ROM-PACK slot on the side of the computer. Unlike the VS-1, however, an external speaker is required, as the SP-1 routes its audio directly through the computer and into your television monitor!

We also have a version of our VOX-EDITOR to let you build your voice messages quickly and easily, and then save them to cassette when you are done. The speech files can later be read by Basic in your application programs! The comprehensive manual we supply gives you all the information you need to make your CoCo talk!

SP-1 pack and manual $179.95
SP-1 manual only (price applied to SP-1 purchase) ..... $ 19.95

TRANSLATOR FOR SP1

Our SC-1/COLOR translator uses the same highly accurate algorithm as our SC-1 for the VS-1, but drives the SP-1 on the color computer. The SC-1/COLOR, however, is even easier to use! When the SC-1/COLOR translator is loaded, it becomes a part of Basic! To translate and speak English text, numeric data (even punctuation if you desire), you simply program PRINT #1 followed by your text. The translator takes it, converts it to a phoneme stream, and drives the SP-1 directly!

SC-1/COLOR cassette and manual (requires SP-1 for operation) $89.95
(manual not available separately)

SSB UTILITIES DISK

Two of the shortcomings of SSB DOS are that, one, with six-character file names and default extensions it's hard to know from a name what is in a file, and, two, with no disk information sector the only way to know what is on a disk is to list the files! Since we couldn't conveniently make the file names longer, we did the next best thing and wrote a family of programs to allow you to 'tame' your disks!

Utilities #1 provides a NAME utility to build a disk information sector on each newly-formatted disk. UPDATE lets you change this sector, LOCK lets you write, delete or even LIST protect the files on the disk. LIST is a better listing utility which reads the disk access code from the information sector and will only list LOCK'ed files if you supply the access code. A number of other programs are also included on the disk.

The manual provides information about the way SSB disks are formatted and how the chained sector method work in addition to complete descriptions about each of the programs provided on the disk!

UTILITIES #1 disk and manual $20.95
(manual not available separately)

FANTASTIC PROOFREADER

We are proud to offer Pete Stark's Spell 'n Fix proofreader. We examined a number of very good pieces of spelling correction software. We compared ease of use, speed, correction capability, dictionary maintenance and expansion , disk space overhead, etc..., and found that Pete Stark of StarKits has done a remarkable job on this program. Look at these features and make your own decision.....

MULTIPLE DICTIONARIES - Spell 'n Fix allows you to create multiple dictionaries. In this way, if you want, you can put technical and special words in a separate dictionary. Then, when you proofread your documents, select the dictionaries which have the words you need, thus speeding up the job!

EXECUTE DOS COMMANDS - Need to list your disk to decide which dictionary or text file to proofread? Want to delete an old file? View a file? Most of your DOS commands can be executed directly from within Spell 'n Fix.

MULTIPLE CORRECTION OPTIONS - With Spell 'n Fix, you can add new words to your dictionary at any time. Or, if you want, you can create a new dictionary for words not found in the existing dictionaries. Or simply print out the new words. Or even ignore them! CORRECTION OR MARKING - With Spell 'n Fix, you can mark incorrectly-spelled words in your text for later correction, or you can tell Spell 'n Fix to correct them as they are found. The choice is yours!

COMPRESSED DICTIONARY - Spell 'n Fix uses either of two forms of dictionary, and converts either to the other at your command. The first type is a fully expanded dictionary with one word per line, allowing you to use your editor to make corrections, deletions and additions easily. The second type incorporates a slick compression method to drastically reduce dictionary size and increase access speed.

After looking at our options, we decided that this was the program for us, and are proud to be the exclusive agents for StarKits for the SSB DOS68 and DOS69 versions. We can also supply versions for 6800 and 6809 Flex. The OS-9 version will be available soon directly from StarKits. StarKits also has a version for the Color Computer! At the price, we feel that no other program can do what Spell 'n Fix can.....

SPELL 'N FIX disk (with 17000 word dictionary) and manual $178.50
(manual not available separately)

8850 SPEECH SYNTHESIZER

The VS-1 is a completely-assembled, tested unit for the SS-50 I/O bus which gives your computer the power of unlimited speech. Since the VS-1 is programmed phonetically, there is essentially no limitation on what you can make your computer say! The VS-1 programs just like a parallel printer driver, and requires an average of only one phoneme per letter in English for a very low message memory storage overhead. The data rate is typically a low 70 baud.

The VS-1 is supplied with the easiest-to-use, most flexible speech editing program available. This program makes building your speech files or data a real pleasure. We also supply additional software and sample speech files on the disk to give you a 'leg up' on making your computer talk.

The manual provides sections on speech generation theory, operation of the supplied software, complete technical descriptions, a sample speech dictionary and phonetic code charts, fifty pages in all!

If you are a roll-your-own nut, we provide the bare board or manual or disk or synthesizer chip, or combinations ! We suggest that only VERY experienced persons attempt such assembly due to the fact that we do not provide assembly instructions beyond parts list parts layout and schematic).

The VS-1 is available for 6800 and 6809 FLEX, 6800 and 6809 SSB DOS, and OS-9 Level I and II.

VS-1 complete with manual and disk $294.95
VS-1 manual and disk only (may be applied to brand-X) ..... $ 29.95
VS-1 manual only (may be applied to VS-1 order. Look before you leap!) ..... $ 19.95
VS-1 bare board (build your own...requires experience!) ..... $ 39.95
SC-01A synthesizer chip ..... $ 60.00
VS-1 bare board, chip, disk and manual ..... $ 99.95

TRANSLATOR FOR VS1

If you have our VS-1 (or even brand-X), and want an easier way to make your computer talk, our SC-1 translator is for you. The SC-1 takes plain English text from a disk file or your keyboard, translates it to a phoneme stream, and sends the resulting output to either the speech synthesizer hardware or to a data file for later use.

The SC-1 translator is one of the most accurate such programs we have seen, making typically only one 'mistake' per sentence, for an average accuracy of about 90%! Due to the type of algorithm we are using, even proper names are generally translated correctly! Where the SC-1 is being used to compile 'canned' messages for later use, we provide an additional program which takes a binary phoneme file created by the SC-1 and converts it to a speech file which may be edited by our VOX-EDITOR (supplied with the VS-2). In this way, you can quickly create your messages and then 'fine-tune' them!

The SC-1 translator is available for 6800 and 6809 FLEX, 6800 and 6809 SSB DOS, and OS-9 level I and II. If you are using brand-X, order the SC-1 with the VOX-EDITOR on the same disk!

SC-1 disk and instructions $39.95
SC-1 disk, instructions with VOX-EDITOR $49.95

TREK-68

Our TREK program is the best 'real-time', 'live-action' game available for 68xx computers using memory-mapped displays. This program does not require graphics capabilities from your display. Otherwise, this is a full arcade-type game. No triviality here, either. The suicide option has only been won by a handful of people world-wide, and can run for hours!

TREK is available for 6800 and 6809 FLEX and 6800 and 6809 SSB DOS, and requires both a memory-mapped display and a source of 60 or 100 Hz interrupts (such as the MP-1 timer board).

TREK disk and manual $24.95

COLOR-TREK

Like our other version of TREK, COLOR-TREK provides you seat-of-the-pants action. This program uses no routines from Basic, and so can run on non-extended Basic machines. 16K memory is required. Only one player to our knowledge (other than the author) has won the suicide option of this game, and it took him fifty-seven minutes! No triviality here!

COLOR-TREK cassette and manual $17.95

'68' MICRO JOURNAL

★ The only ALL 6800 Computer Magazine.

★ **More** 6800 material than all the others combined:

MAGAZINE COMPARISON
(2 years)
Monthly Averages
6800 Articles

KB	BYTE	CC	DOBB'S	TOTAL PAGES
7.8	6.4	2.7	2.2	19.1 ea. mo.

Average cost for all four each month: $6.53
(Based on advertised 1-year subscription price)

'68' cost per month: $2.04

That's Right! Much, Much More
for About
1/3 the Cost!

OK. PLEASE ENTER MY SUBSCRIPTION

Bill My: Master Charge ☐ — VISA ☐

Card # ________ Exp. Date ________

For ☐ 1-Year ☐ 2 Years ☐ 3 Years

Enclosed: $________

Name________

Street________

City________ State________ Zip________

My Computer Is:________

68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

SUBSCRIPTION RATES

USA
1 Year $24.50, 2 Year $42.50, 3 Year $64.50

*FOREIGN SURFACE Add $12.00 per Year to USA Price

*FOREIGN AIRMAIL Add $36.00 per Year to USA Price

**CANADA & MEXICO Add $5.50 per Year to USA Price
Cash (USA) or drawn on a USA Bank!!!

Universal Data Research, Inc. Introduces

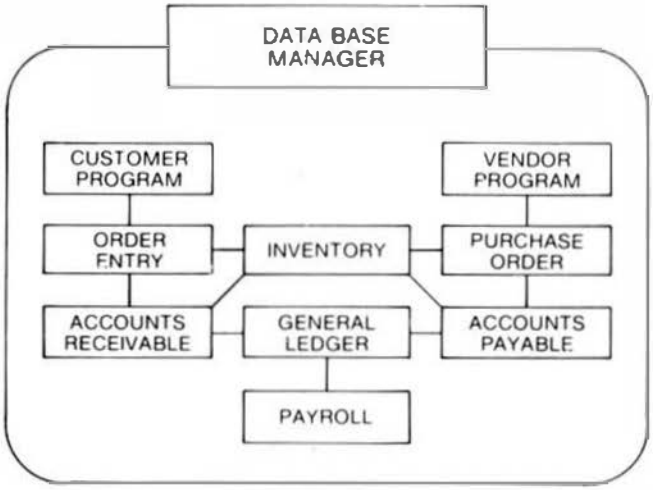
MODEMS

300 Baud Acoustic	$149.00
300 Baud Direct	$179.00
300 Baud Auto Answer	$219.00
1200 Baud Direct	$449.00
1200 Baud Auto Answer	$499.00
300/1200 Auto Answer	$549.00
300/1200 Auto Dial	$599.00
Modem 2 Port I/O Card	$119.00
with speed select	
Printer Special: Okidata 82A	$449

Dealer Inquiries Welcome ● Call or Write For Free Catalogue

New, Lower Prices

FLEX* and UniFLEX*
software for the 6809



Integrated Business Programs

	FLEX	UniFLEX
Accounts Payable	$295	$395
Accounts Receivable	$295	$395
General Ledger	$295	$395
Inventory 2	$295	$395
Payroll	$295	$395
Data Base Manager	$350	$450
Word Processing		
Software	$295	$395
WP Menu		$150
P Control		$150

UNIVERSAL DATA RESEARCH INC.

*FLEX & UniFLEX are Trademarks of Technical Systems

2457 Wehrle Drive, D-68, Buffalo, NY 14221
PHONE (716) 631-3011

Dealer Inquires Welcome

master charge VISA

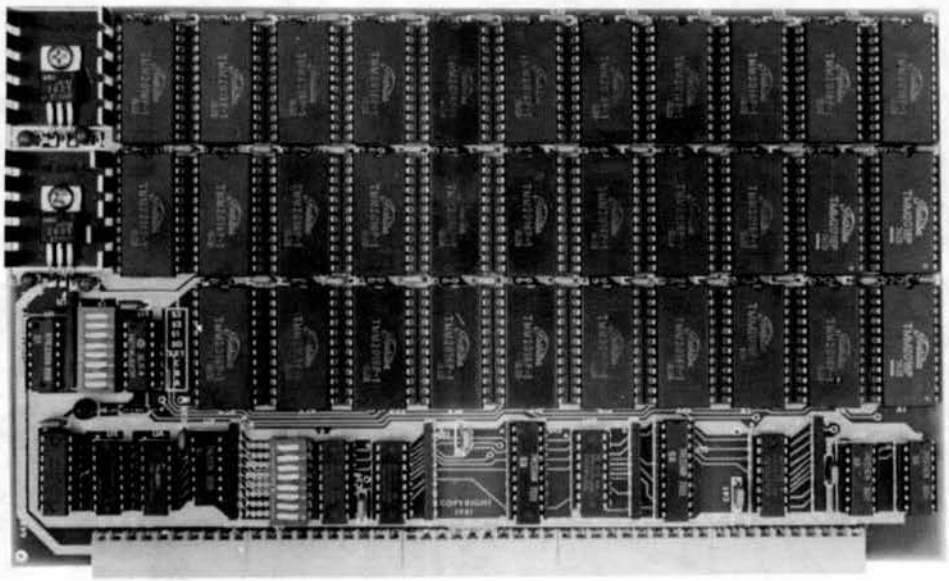
64K SS-50 STATIC RAM

NEW!

$179.00 (48K KIT)

NEW!

LOW POWER!



RAM OR EPROM!

BLANK PC BOARD WITH DOCUMENTATION $52

SUPPORT ICs + CAPS -	$18.00
FULL SOCKET SET -	$15.00

ASSEMBLED AND TESTED **ADD $40**

FEATURES:

- ★ Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- ★ Fully supports Extended Addressing.
- ★ 64K draws only approximately 500 MA.
- ★ 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- ★ Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- ★ 2716 EPROMs may be installed anywhere on Board.
- ★ Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- ★ One Board supports both RAM and EPROM.
- ★ RAM supports 2MHZ operation at no extra charge!
- ★ Board may be partially populated in 16K increments.

56K	$210
64K	$249

16K STATIC RAMS?

The new 2K x 8, 24 PIN, static RAMs are the next generation of high density, high speed, low power, RAMs. Pioneered by such companies as HITACHI and TOSHIBA, and soon to be second sourced by most major U.S. manufacturers, these ultra low power parts, feature 2716 compatible pin out. Thus fully interchangeable ROM/RAM boards are at last a reality, and you get BLINDING speed and LOW power thrown in for virtually nothing.

Digital Research Computers
(OF TEXAS)
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $2.00 postage. We pay balance. Order under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for insurance.

ARCADE 50

POWERFUL COLOR GRAPHICS
Uses the new TMS9918A Video Display Processor. High resolution 256 x 192 pixel display with 15 colors.16k Bytes of onboard RAM does not reduce user memory. 32 graphic images can be individually moved with simple X-Y commands for smooth animation.
External Video input allows subtitling.
NTSC composite video output.
SOUND EFFECTS AND MUSIC
Three AY3-8910 Programmable Sound Generators
Nine simultaneous voices
Three independent noise sources
Onboard stereo amplifier drives two 8 ohm speakers
ADDITIONAL I/O CAPABILITIES
Eight analog inputs with 8 bit resolution
Supports four joysticks with pushbutton switches
Eight bit parallel I/O port
Entire unit maps into 256 bytes of memory
DOCUMENTATION AND SOFTWARE
Programming manuals for Video and Sound Processors
Subroutine library and Super Demo Maze Game
Example programs in BASIC, FBASIC and ASSEMBLY
User library and sales support

ARCADE 50 assembled and tested	$325.00
Video and Audio connector set	15.00
4 Joystick connector set	15.00
2 Radio Shack Joysticks	24.00
UHF channel 33 modulator	32.00
Gold Molex connectors	12.00
A/BASIC for 6800	110.00
FBASIC for 6809	110.00
FBASIC (with ARCADE 50)	75.00
FBASIC (manual only)	10.00
ARCADE 80 (TRS Model I)	395.00
ARCADE 100 (S-100 BUSS)	375.00
ARCADE 50 RGB	375.00
LABVIDEO (Motorola EXORbus)	375.00
LABVIDEO RGB	375.00
NEW MV09 6809 Processor Board	225.00
*Comes assembled with PIA and ACIA	
*12 Sockets for 2716, 2732 or RAM	
*Supports DMA disk I/O	
*Ideal for 6809 upgrade or process control	
AMDEK COLOR I Monitor	425.00
AMDEK COLOR II Monitor	799.00
AMDEK COLOR III Monitor	499.00
256K Dynamic Memory Board (assembled)	795.00
256K Dynamic Memory Board (assembled w/64K)	395.00
64K Dynamic Memory Board (assembled)	295.00

Specify 5" or 8" soft sector disk for TSC's FLEX or MICROWARE'S OS/9 system
TERMS: CASH, VISA, MC, C.O.D.

FBASIC

TERMINUS DESIGN INC. in conjunction with Microware Systems Corporation, is proud to announce FBASIC-an enhancement of Microware's 6800 A/BASIC. The fast compiled BASIC has been adapted for 6809 users with added video and sound features for ARCADE 50 users. FBASIC is a true compiler that produces optimized machine language modules which are ROMable and require no Run-Time package. FBASIC requires less memory overhead and runs hundreds of times faster than BASIC interpreters. It supports standard BASIC instruction including String functions, Disk I/O and fast integer arithmetic with multiple-precision capability. Graphics verbs and functions fully support the Arcade 50. Arcade statements include:

INIT	MODE	BLANK	BACKDROP
SIZE	MAG	VREG	DELAY
MOVE	DRAW	FCOLOR	JSWITCH
REMOVE	RDRAW	BCOLOR	SWITCH
PSG	TONE	ENVL	VOLUME
ADC	SPRITE	SPNAME	ENDEF
SPCOLOR	RSPRITE	SPOEF	PATDEF
VPEEK	VPOKE	VPRINT	

TERMINUS DESIGN INC
16 SCARBROUGH ROAD
ELLENWOOD, GA 30049
(404) 474-4855

68 MICRO JOURNAL PROGRAMS on DISK

Disk #1: FILESORT, MINICAT, MINICOPY, MINIFMS, **LIFETIME, **POETRY, **FOODLIST, **DIET.
Disk #2: DISKEDIT w/ Inst. & fixes, PRIME, *PRMOD, **SNOOPY, **FOOTBALL, **HEXPAWN, **LIFETIME.
Disk #3: CBUG09, SEC1, SEC2, FIND, TABLE2, INTEXT, DISK-EXP, *DISKSAVE.
Disk #4: MAILING PROGRAM, *FINDDAT, *CHANGE, *TESTDISK.
Disk #5: *DISKFIX 1, *DISKFIX 2, **LETTER, **LOVESIGN, **BLACKJAK, **BOWLING.
Disk #6: **PURCHASE ORDER, INDEX (Disk file indx).
Disk #7: Linking Loader & RLOAD, Harkness
Disk #8: CRTSET, Lanpher (May '82)
Disk #9: DATECOPY, DISKFIX9 (Aug '82)

NOTE: All are as published or received by 68 Micro Journal, some have fixes and patches.

This is a reader service only! No Warranty is offered or implied, they are as received and are for reader convenience ONLY. Also 6800 and 6809 programs are mixed, as each is fairly simple (mostly) to convert to the other.

PRICE: 8" Disk $19.95 - 5" Disk $17.95

68 MICRO JOURNAL
POB 794
Hixson, TN 37343
615-842-4600

* indicates 6800, ** indicates BASIC SWTPC or TSC - 6809 no indicator.

MASTER CARD - VISA accepted - Foreign add sufficient postage surface or air!!

WINCHESTER FOR MOTOROLA EXORCISOR/MDOS

☐ 10 MB Winchester hard disk runs MDOS on Motorola Exorcisor System ☐ No modification to MDOS required ☐ MDOS based software stays alive ☐ All user software operates without modification ☐ Optional SA-801R flexible diskette drive system ☐ Optional 10 MB removable cartridge.

CSA *For information call (619) 566-3911*
Computer System Associates
7562 Trade Street, San Diego, CA 92121

TEN MOST-ASKED QUESTIONS

ABOUT DYNACALC™

THE ELECTRONIC SPREAD-SHEET FOR 6809 COMPUTERS

1. **What is an electronic spread-sheet, anyway?**
   Business people use spread-sheets to organize columns and rows of figures. DYNACALC simulates the operation of a spread-sheet without the mess of paper and pencil. Of course, corrections and changes are a snap. Changing any entered value causes the whole spread-sheet to be re-calculated based on the new constants. This means that you can play, 'what if?' to your heart's content.

2. **Is DYNACALC just for accountants, then?**
   Not at all. DYNACALC can be used for just about any type of job. Not only numbers, but alphanumeric messages can be handled. Engineers and other technical users will love DYNACALC's sixteen-digit math and built-in scientific functions. There's even a built-in sort command, so you could use DYNACALC to manage small data bases - up to 256 records.

3. **What will DYNACALC do for ME?**
   That's a good question. Basically the answer is that DYNACALC will let your computer do just about anything you can imagine. Ask your friends who have VisiCalc, or a similar program, just how useful an electronic spread-sheet program can be for all types of household, business, engineering, and scientific applications.

4. **Do I have to learn computer progremming?**
   NO! DYNACALC is designed to be used by non-programmers, but even a Ph.D. inComputer Science can understand it. Built-in HELP messages are provided for quick reference to operating instructions.

5. **Do I have to modify my syatem to use DYNACALC?**
   Nope. DYNACALC uses any standard 6809 configuration, so you don't have to spend money on another CPU board or waste time learning another operating system.

6. **Will DYNACALC read my existing data files?**
   You bet! DYNACALC has a beautifully simple method of reading and writing data files, so you can communicate both ways with other programs on your system, such as the Text Editor, Text Processor, Sort/Merge, RMS data base system, or other programs written in BASIC, C, PASCAL, FORTRAN, and so on.

7. **How fast is DYNACALC?**
   Very. Except for a few seldom-used commands, DYNACALC is memory-resident, so there is little disk I/O to slow things down. The whole data array (worksheet) is in memory, so access to any point is instantaneous. DYNACALC is 100% 6809 machine code for blistering speed.

8. **Is there a veraion of DYNACALC for MY system?**
   Probably. You need a 6809 computer (32k minimum) with FLEX or UniFLEX operating system. A version for OS-9 is also in the works. You also need a decent CRT terminal, one with at least 80 characters per line, and direct cursor addressing. If your terminal isn't smart enough for DYNACALC, you probably need a new one anyway. The UniFLEX version of DYNACALC also allows you to mix different brands of terminal on the same system. There's also a special version of DYNACALC for Color Computers equipped with FLEX and Data-Comp's F-MATE. A version for Frank Hogg's Color Computer FLEX is also being done.

9. **How much does DYNACALC coat?**
   The FLEX versions are just $200 per copy; UniFLEX version $395. Foreign orders add $10 per copy for postage. We encourage dealers to handle DYNACALC, since it's a product that sells instantly upon demonstration. Call or write on your company letterhead for more information.

10. **Where do I order DYNACALC?**
    See your local DYNACALC dealer, or order directly from CSC at the address below. We accept telephone orders from 10 a.m. to 6 p.m., Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Please specify diskette size for FLEX versions. Software serial number is required for the UniFLEX version of DYNACALC.

ORDER YOUR DYNACALC™ TODAY

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 7A Burton St., Darlinghurst, NSW 2010 Sydney. Telephone: 02-357-5111.

United Kingdom: order from Compusense. Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

Computer Syatema Center
13461 Olive Blvd.
Chesterfield, MO 63017
(314) 576-5020

OS-9 Version NOW Available

$250.00

UniFLEX software prices include maintenance for the first year.

DYNACALC, DYNAMITE, and DYNAMITE + are trademarks of Computer Systems Center.

F-MATE is a trademark of Data-Comp
VisiCalc is a trademark of VisiCorp
OS-9 is a trademark of Microware and Motorola
FLEX and UniFLEX are trademarks of TSC

ALSO FROM CSC

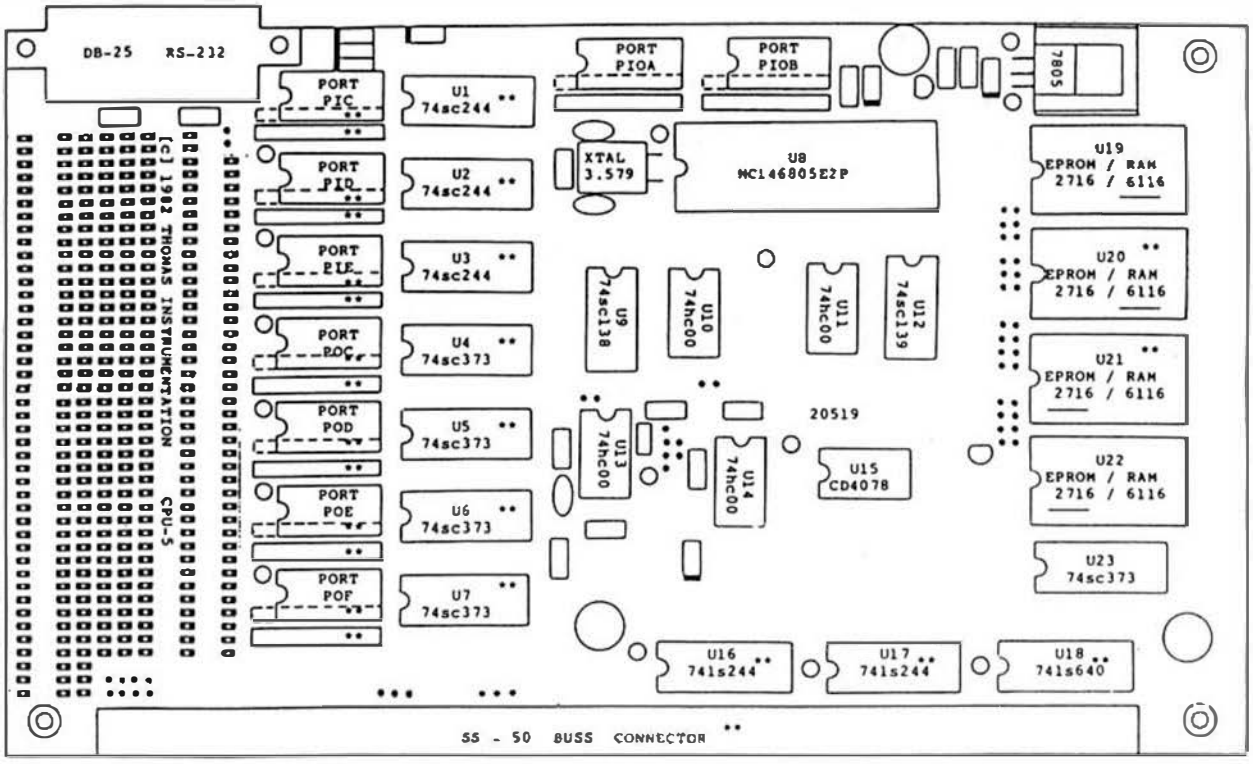
DYNAMITE +
"THE CODE BUSTER"
now available for UniFLEX
OS-9 version soon

DYNAMITE + is a new version of DYNAMITE, our popular 6809/6800 disassembler package for 6809 FLEX. Present users of DYNAMITE can upgrade to DYNAMITE + by sending us the original DYNAMITE diskette and $40 (plus $5 for foreign postage). DYNAMITE + does everything DYNAMITE does, and more! A cross-reference generator has been added, label files are now maintained only in text form (LABEL EQU $xxxx), and boundary file specifications have been tremendously simplified, which makes it easier to disassemble large programs containing lots of big tables.

The UniFLEX version of DYNAMITE + does everything the FLEX version does, and also automatically handles system calls and 'info' areas.

DYNAMITE + is available for $100 per copy on FLEX (specify diskette size), and $300 on UniFLEX. Foreign orders add $5 per copy for postage.

*** NEW 6805 ALL CMOS CARD ***



CPU-5-B	Blank card with documentation	$49.00	
CPU-5-P	Partial card (parts marked ** not incl.)	$229.00	Includes EPROM Monitor
CPU-5-F	Full card with all parts shown (fully assembled)	$349.00	and 115 vac Power Pak.
CROSS-ASSEMBLERS	for FLEX ([c] TSC) 6809 system	$150.00	

SOFTWARE

VDISK Treat extended memory as a super fast disk drive

6809 source & object	$149.00
6809 object	$ 99.00

OUTSIDE MODEM PROGRAM Incl. source

UniFLEX version	$100.00
FLEX version 6800 & 6809	$ 50.00
HAYES SMART MODEM	$249.00

CROSS ASSEMBLERS for 6800,6801,6805 runs on 6809 FLEX $150.00

TV-EDIT Screen oriented Editor for FLEX 6809 $ 95.00
Please specify 5" or 8" Disk when placing order.

CHIPS

Please note when ordering IC's that a $100.00 Minimum is in effect Orders less than the minimum will be charged an additional $10.00 for handling

6116P-3	2MHZ	$13.50
2016P-2	2MHZ	$10.50
2716	1MHZ	$ 6.50
2716-1	2MHZ	$16.50
146805E2P		$25.00
6802		$10.00
6809		$25.00
6821		$ 6.50
6840		$12.00
6850		$ 6.75
74LS640		$ 3.25

HARDWARE

NEW MEMORY PRICES TO INTRODUCE VDISK

S-R/R without memory chips	$120.00
with 8K NMOS@2MHZ	$149.00
with 16K NMOS@2MHZ	$199.00
with 32K NMOS@2MHZ	$299.00
with 48K NMOS@2MHZ	$399.00
Bare Card	$ 49.00

Extender Cards assemb with logic aid

SS-30/50C	$ 35.00
SS-30	$ 25.00
SP-1 Prototype Board A/T	$195.00
SP-1 Bare Card	$ 49.00
SS-50 Wire Wrap Board B/C	$ 39.00
SS-30 Wire Wrap Board B/C	$ 20.00
6802 Super CPU A/T	$235.00
6802 Super CPU B/C	$ 39.00
Video RAM, B/C $49.00, A/T	$195.00
Parallel I/O, B/C $49.00, A/T	$139.00
SS-50 Backplanes 4.6.8.12.&14 position @ $5.00 per slot w/o connectors	
SS-30 8 pos. BP w/o connect.	$ 39.00
Transition Card $49.00, A/T	$ 95.00
Molex Gold Male $1.60, Female	$ 1.60
Molex Tin Male $ .40, Female	$ .50

'68' MICRO JOURNAL ADVERTISERS INDEX

This Index is provided as a reader service. The publisher does not assume any liability for omissions or errors.

TO REALIZE THE FULL POWER & PERFORMANCE OF THE 6809, LOOK TO GIMIX.

GIMIX OFFERS YOU A VARIETY OF SS50 BUS COMPONENTS AND SYSTEMS.

OS-9 GMX III

The GMX 6809 CPU III and OS-9 GMX III. A Multi-user, Multi-tasking package for the ultimate in System Performance plus protection of the system and other users from crashes caused by errors in individual users programs.

#01 (CPU & Software) ..... NEW! **$1695.01**

INTELLIGENT I/O PROCESSOR BOARDS increase system throughput by reducing interrupts to the host, buffering data transfers, and data preprocessing. Prices include on board firmware. Requires system drivers.

#11 3 port RS232 Serial (SS30) ..... NEW! **$498.11**
#12 4 port Parallel (SS50) **$538.12**
OS-9 GMX III drivers . . (included when purchased with GMX III package) . . **$200.00**
OS-9 Level 2 users -- contact GIMIX for system requirements and availability.

192K GMX III #79 SYSTEMS: All include GMX 6809 CPU III and OS-9 GMX III (#01); a #11 3 port intelligent serial I/O & cables; #19 Classy Chassis; 192KB Static RAM, #68 DMA controller, all necessary cables, power regulators, and filler plates. The OS-9 Editor, Assembler, Debugger, BASIC-09, and RUNB are included.

#79 with dual 40 track DSDD drives **$5998.79**
#79 with dual 80 track DSDD drives ..... NEW! **$6298.79**
#79 with **#88** 8" Dual Drive Disk System **$7598.79**
#79 with **#90** 19MB Winchester subsystem & one 80 track DSDD drive . . **$8998.79**
UniFLEX for the GMX 6809 CPU III and Intelligent I/O boards is in development.

OS-9 GMX I; OS-9 GMX II; FLEX; and UniFLEX

The **#05** GIMIX 6809 PLUS CPU board **$578.05**
Options: GMX OAT ..... $35.00 SWTPC OAT $15.00
9511A ..... $312.00 9512 ..... $265.00

#49 64KB GHOST SYSTEM includes: #05 CPU; #19 Classy Chassis; 64KB static RAM; a #43 2 port serial card & cables; #68 DMA Controller; all necessary cables, power regulators, and filler plates; GMXBUG monitor; FLEX; and OS-9 GMX I. You can software select either FLEX or OS-9. The OS-9 Editor, Assembler, Debugger, BASIC-09, and RUNB are also included.

#49 with dual 40 track DSDD drives **$4398.49**
#49 with dual 80 track DSDD drives **$4698.49**
#49 with **#88** 8" Dual Drive Disk System **$5998.49**
#49 with **#90** 19MB Winchester subsystems & one 80 track DSDD drive **$7398.49**

#39 128KB SYSTEM includes: #05 CPUwOAT; #19 Classy Chassis; 128KB of static RAM; a #43 2 port serial card & cables; #68 DMA Controller; all necessary cables, power regulators, and filler plates; GMXBUG monitor; FLEX; and OS-9 GMX II. You can software select either FLEX or OS-9. The OS-9 Editor, Assembler, Debugger, BASIC-09, and RUNB, and GMX-VDISK for FLEX are included.

#39 with dual 40 track DSDD drives **$4998.39**
#39 with dual 80 track DSDD drives **$5298.39**
#39 with **#88** 8" Dual Drive Disk System **$6598.39**
#39 with **#90** 19MB Winchester subsystem & one 80 track DSDD drive **$7998.39**
UniFLEX, available at extra cost, requires 8" or Winchester drives. A signed license agreement with TSC is required before shipment.

You can add to any GIMIX system RAM, I/Os and other options, or substitute non-volatile RAM. GIMIX will customize to your needs.

COMING SOON: Contact GIMIX for price and availability on 40MB and 72MB Winchester (5¼") drives, removeable pack Winchesters, 256KB static RAM boards.

All GIMIX systems are guaranteed for 2MHz operation. GIMIX systems include documentation for all boards and software in a GIMIX binder. **ALL DRIVES ARE 100% TESTED AND ALIGNED BY GIMIX.**

ALL BOARDS AND SYSTEMS ARE ASSEMBLED, BURNED-IN, AND TESTED. GOLD-PLATED BUS CONNECTORS ARE USED.

TO ORDER BY MAIL: SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear. U.S. orders add $5 handling if order is under $200.00. Foreign orders add $10 handling if order is under $200.00. Foreign orders over $200.00 will be shipped via Emery Air Freight COLLECT, and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, 231 S. LaSalle Street, Chicago, IL 60693, account #73-32033. Visa or Master Charge also accepted.

EXPORT MODELS: ADD $30 FOR 50Hz. POWER SUPPLIES.

GIMIX Inc. reserves the right to change pricing, terms, and product specifications at any time without further notice.

ALL PRICES ARE F.O.B. CHICAGO

Choose from GIMIX' wide variety of system components.

The GIMIX CLASSY CHASSIS #19 consists of a heavyweight aluminum cabinet, constant voltage ferro-resonant power supply, and SS50 Mother board with baud rate generator board **$1398.19**

Triple Disk regulator card and cables ..... $88.22 Baud rate generator card $88.93
Missing cycle detector ..... $38.23 Filler plates $14.92
Back panel connector plates (specify) . $8.60 50 Hz. option $30.00

MEMORIES (GIMIX uses only Static RAM)

#67 64KB NMOS STATIC RAM board **$478.67**
#64 64KB CMOS STATIC RAM board w/battery back-up **$568.64**
#34 8K PROM board **$98.34**
#32 16 socket PROM/ROM/RAM board **$238.32**

I/O Boards (see above for Intelligent I/Os)

#41 Single port serial, RS232/20ma. current loop **$88.41**
#43 2 port serial, RS232 ..... **$128.43**
#46 8 port serial, RS232 ..... **$318.46**
#42 2 port parallel **$86.42**
#45 8 port parallel **$198.45**
#50 serial, RS232, RS422, RS423 ..... **$244.50**
#52 SSDA serial, RS232, RS422, RS423 ..... **$254.52**
#54 ADLC serial, RS232, RS422, RS423 ..... **$268.54**
Each cable with connectors for back panel mounting (specify board) **$24.95**

DISK CONTROLLERS

#68 DMA (featured in all systems above) **$588.68**
#28 dbl. dens. programmed I/O (5" drives only) **$298.28**
#58 single dens. programmed I/O (5" and/or 8" drives) **$226.58**
#48 same as **#58** but for 5" drives only **$198.48**
Cable sets: 8" with Back Panel connector **$29.25**
for two 8" external drives **$44.26**
for two 5" drives **$34.96**

SOFTWARE: GIMIX exclusive versions of OS-9/GMX I, II, III & FLEX are for GIMIX hardware only. All versions of OS-9 require the #68 controller.
When ordered with any controller, FLEX is **$30.00**

GMXBUG PROMs and manual **$98.85**
Boot or Video boot PROM ..... $30.00 UNIFLEX boot PROM **$50.00**
OS-9 GMX I ..... $200.00 OS-9 GMX II **$500.00**
Editor ..... $125.00 Assembler **$125.00**
BASIC-09 ..... $200.00 RUNB **$100.00**

DISK DRIVES FOR GIMIX SYSTEMS – complete with cables and power regulators.

5" DSDD 40 track ..... 2 for $900.00
5" DSDD 80 track ..... 2 for $1300.00
#88 8" Dual 8" DSDD drives, cabinet, power supply, & cables **$2698.88**
Cabinet only ..... $848.18 220V 50Hz. Option, add **$30.00**
Filler plate ..... $14.83 Cable for 2 drives **$44.82**
Cable for 4 drives ..... $57.84 Cable for cabinet to mainframe **$45.81**

WINCHESTER SUBSYSTEMS: for use only in GIMIX systems with **#68** DMA controller.

#90: includes one 19MB drive, interface, and Software **$3688.90**
#91: includes two 19MB drives, interface and Software **$5288.91**
Contact GIMIX for price and availability of other forthcoming subsystems.

OTHER BOARDS

#75 GHOST 80X24 VIDEO BOARD **$398.76**
#66 50 pin Proteboards ..... $56.66 **#33** 30 pin Protoboards **$38.33**
#03 6800 CPU **$224.03**
#05 6800 CPU with timers ..... $288.06 Baud rate option, add **$30.00**

#08 RELAY DRIVER (board, bracket, transformer, and 31 relays) **$1128.08**
#86 - #08 (board, bracket, transformer, without relays) **$538.86**
#85 OPTO board **$348.85**
WINDRUSH EPROM PROGRAMMER **$375.00**
3" Binder ..... 12.00 2" Binder **$9.00**

GIMIX DOES NOT GUARANTEE PERFORMANCE OF ANY GIMIX SYSTEMS, BOARDS OR SOFTWARE WHEN USED WITH OTHER MANUFACTURERS PRODUCT.

DON'T SEE IT??? ASK! OUR BROCHURE HAS MORE COMPLETE DESCRIPTIONS AND SPECS. PHONE OR WRITE TODAY FOR YOUR COPY.

GIMIX INC. 1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609
(312) 927-5510 • TWX 910-221-4055

The Original FLEX™ for Color Computers

- Upgrade to 64K
- RS to FLEX, FLEX to RS file transfer ability
- Create your own character set
- Automatic recognition of single or double density and single or doubled sided
- All features available for either single or multiple drive systems
- Sellable Disk Drive Seek Rates
- Faster High Resolution Video Display with 5 different formats
- Save RS Basic from RAM to Disk
- Move RS Basic to RAM
- Load and save function on FLEX disk
- 24 Support Commands 12 with Source Text
- External Terminal Program

Languages Available

Pascal, Fortran, RS Basic, RS Assembler, TSC Basic, TSC Assembler, Relocating Assembler, Macro Assembler, Mumps

If you are tired of playing games on your TRS-80C Color Computer, or find that you are handicapped by the limitations of the RS BASIC in trying to write a Program that will allow you to actually USE the Color Computer as a COMPUTER, YOU ARE READY TO MOVE UP TO THE **FLEX9**™ Operating System. If you want to have REAL PROGRAMMING POWER, using an Extremely Powerful Business BASIC, PASCALs, C Compilers, a full-blown Macro Assembler with a Library capability so you are not continuously reinventing the wheel, YOU ARE READY TO MOVE UP TO THE **FLEX9** Operating System. If you would like to see if YOU REALLY COULD USE A COMPUTER IN YOUR BUSINESS, or begin to make your Computer start PAYING IT'S OWN WAY by doing some Computer Work for the millions of small businesses around you, such as Wordprocessing, Payroll, Accounting, Inventory, etc., then YOU ARE READY TO MOVE UP TO THE **FLEX9** Operating System. How?? **DATA-COMP** has the way!

DATA-COMP's FLEX9 Conversion for the TRS-80C Color Computer was designed for the SERIOUS COMPUTER **USER**, with features like greatly increased Display Screens WITH Lower Case Letters, so you can put a FULL Menu on ONE Screen, or see SEVERAL Paragraphs at the same time, with features like providing a FULL Keyboard so you have FULL Control of your Computer AND it's Programs NATURALLY, without needing a chart to see what Key Combination will give you what function, with USER ORIENTED functions to make using the Operating System natural, like having the Computer AUTOMATICALLY determine what type of Disk is being used in what type of Disk Drive and working accordingly, rather that you have to specify each and every thing for it, or like having the Computer work with the Printer you have been using all along without you having to tell the new Operating System what is there, etc.

™FLEX is a trademark of Technical Systems Consultants

DATA-COMP has everything you need to make your TRS-80C Color Computer WORK for YOU, from Parts and Pieces to Full, Ready To Use SYSTEMS. DATA-COMP designs, sells, services, and SUPPORTS Computer SYSTEMS, not just Software. CALL DATA-COMP TODAY to make your Computer WORK FOR YOU!

System Requirements

FLEX9 Special General Version + Editor & Assembler (which normally sell for $50.00 ea.)	$150.00
F-MATE(RS) FLEX9 Conversion Rout. for the RS Disk Controller	
when purchased with Special General FLEX9 Sys	$49.95
when purchased without the General FLEX9 Sys	$59.95
Set of Eight 64K RAM Chips w Mod. Instructions	$59.95
Color Computer with 64K RAM and EXT. BASIC	$399.95

SPECIAL SYSTEM PACKAGES

64K Radio Shack COLOR COMPUTER, Radio Shack COLOR DISK CONTROLLER, a Disk Drive System, Special General version of FLEX9™, F-MATE(RS)™ and a Box of 10 Double Density Diskettes, a COMPLETE, ready to run SYSTEM on your Color TV Set. $1079.95

DISK DRIVE PACKAGES, etc.

These Packages include the Radio Shack Disk Controller, Disk Drives with Power Supply and Cabinet, and Disk Drive Cable.

PAK #1 — 1 Single Sided, Double Density Sys	$489.95
PAK #2 — 2 Single Sided, Double Density Sys	$749.95
PAK #3 — 1 Double Sided, Double Density Sys	$569.95
PAK #4 — 2 Double Sided, Double Density Sys	$919.95
PAK #5 — 2 Qume Thinline Double Sided Double Density Sys	$749.95

PARTS AND PIECES

Radio Shack Disk Controller	$179.95
1 Single Sided, Double Density Disk Drive Tandon	$249.95
1 Double Sided, Double Density Disk Drive Qume	$349.95
1 Qume Thinline Double Sided, Double Density	$279.95
Screen Clean — Clears Up Video Distortion On Your Color Computer	$39.95
Single Drive Cabinet with Power Supply	$89.95
Double Drive Cabinet with Power Supply	$109.95
Single Drive Disk Cable for RS Controller	$24.95
Double Drive Disk Cable for RS Controller	$34.95
Micro Tech Prods., Inc. LOWER CASE ROM Adapter	$74.95
Radio Shack BASIC Version 1.1 ROM	$34.95
Radio Shack Extended Basic ROM	$89.95

TERM — AN External Terminal Driver is FREE with purchase of F-Mate

SOFTWARE

CHESS 6809

Requires FLEX™ and one of the following CRT terminals

Now Runs On Any Type Terminal

Features:

- **Two display boards.**
- **Four levels of play.**
- **Point scoring system.**
- **Play white or black.**
- **Change or set-up boards/piece positions.**
- **Forfeit move.**
- **Swap sides.**
- **Make move and swap sides.**
- **Change skill level.**
- **Stop and restart game.**
- **Solve 'Mate in 1-2-3-4' moves.**

$79.95 Specify 5" or 8" disk

This is one of the strongest CHESS programs running on any microcomputer, estimated USCF Rating 1600 + .

Note: Personal checks allow 3-4 week delivery.

DIET-TRAC Forecaster

A Diet Planning and Analysis Program

DIET-TRAC Forecaster is a program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C.P.F. %) or grams of Carbohydrate, Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual.

Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individuals are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. When a weight goal is given (either gain or loss), and a calorie plan is agreed upon between the computer and the individual, the number of days to reach the weight goal is projected. The starting and ending rate of weight loss is calculated, and a daily calendar with each day's predicted weight for a 30-day period is printed.

FLEX VERSION	**$59.95**
UniFLEX VERSION	**$89.95**

PRINTERS

The Epson MX-80

$495.00

The Epson MX-100

$725.00

MX-70 $355.00 MX-80 FT $575.00

☆

Color Computer External Terminal Program

$19.95

☆

Color Computer Stylograph Running Under F-Mate

$195.00

MasterCard VISA

DATA-COMP

P.O. Box 794 *HIXSON, TN 37343*

1-615-842-4601

Verbatim Diskettes.

5" Soft Sector Disks

Single Side Single Density	$2.75 ea.
Single Side Double Density	$2.75 ea.
Double Side Double Density	$4.92 ea.
Plastic Storage Box	$2.00 ea.

8" Soft Sector Disks

Single Side Single Density	$3.75 ea.
Single Side Double Density	$4.10 ea.
Double Side Double Density	$4.75 ea.
Plastic Library Box	$5.00 ea.

Foreign Orders Add 10% Surface—20% Air Mail

WINCHESTER BACKUP UTILITIES

The following utilities allow the backup of any size disk system to any size diskettes.

By simply inserting diskettes as requested by COPYMULT, a large disk system (Winchester, etc.) may be downloaded to your present floppy disk system, any size. No need to fiddle with directory deletions or any of the other tedious operations that must be done using a normal copy routine.

COPYMULT-CMD understands normal "copy" syntax and always keeps up with files already copied by maintaining directories for both host and receiving disk system, thus eliminating hours of tedious keyboard entries and other time consuming cleanup chores.

BACKUP-CMD is a special program that downloads "random" type files, any size.

RESTORE-CMD a special program to restructure copied "random" files for copying, or recopying back to the host system.

FREELINK-CMD a "bonus" utility that "relinks" the free chain of a floppy or hard disk thereby eliminating fragmentation.

** Completely documented source files included.

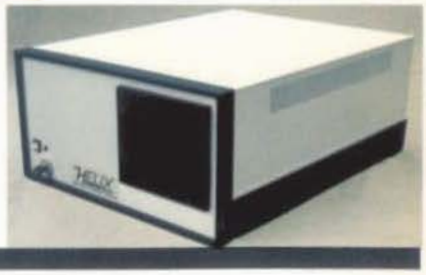
** ALL 3 Programs $99.50 on 8" diskette

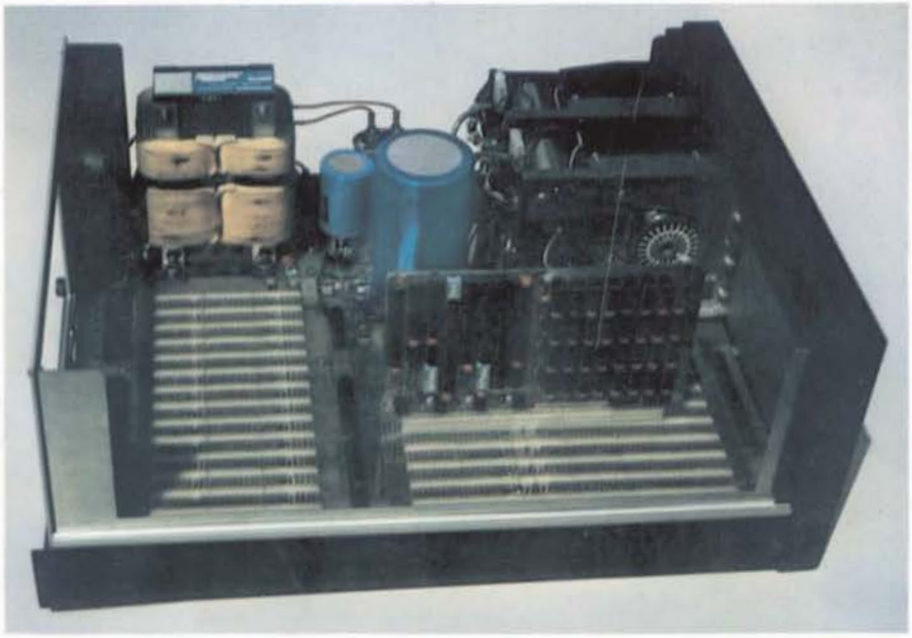
Data-Comp—South East Media & 68 Micro Journal Are Divisions of CPI

68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025

HELIX™





THE MAINFRAME

- Industry Standard Optima™ Cabinet
- Largest Constant Voltage Power Supply in the industry
- S-64 Bus gives 16 Bit Power and S-50 Bus Compatibility
- 10 Main (S-64) Slots
- 14 I/O (S-30) Slots plus 2 On-board
- On-board Baud Rate Generator to 38.4Kb
- Space and Power for two 5¼" Disk Drives
- Full Address Decoding for I/O Slots
- Two RS-232 Serial and Two parallel Ports On-board
- Single Board Construction for Reliability
- Faraday Shielded Bus Lines give "Text Book Clean" Signals

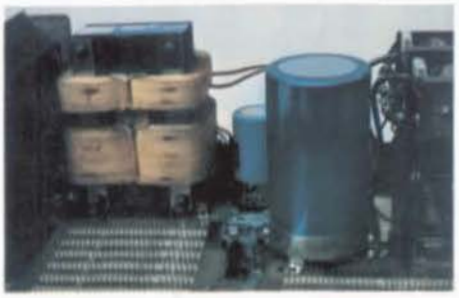
THE PROCESSORS

6809

- Standard 2 MHz Operation
- Standard DAT Compatible with GIMIX and SWTPC
- Standard 6840 Interval Timer
- Standard 1K Scratchpad RAM
- Standard Clock/Calendar with Battery
- Provision for Programmers Console

68000

- Standard 8 MHz Operation
- Memory Management Hardware
- Provision for Programmers Console
- 16 Bit Power and 8 Bit Compatibility



The HELIX™ computer system represents the latest advance in S-50 bus computer systems. Relying on the physical nature of S-50 bus connectors to guarantee compatibility, the HELIX adds 14 bus lines (becoming S-64) to allow a 68000 processor to operate with full 16 bit data transfer and 24 bit addressing, while at the same time providing full interchangability with existing S-50 components.

Offered with a selection of processors, memories, and peripheral controllers, a HELIX system can be configured for applications ranging from advanced hobbyist to multiterminal time-sharing.

Designed to offer the utmost in speed, reliability, and utility at a reasonable price, it represents a new standard of quality for those who require a professionally designed computer for professional use.

THE POWER SUPPLY

- Ferro-resonant Transformer for Line Noise and Under-Voltage Protection
- Conservative 25 Amps at 8.5 Volts
- Conservative 5 Amps at ±16 Volts
- Conservative Component Rating for Reliability

THE COMPONENTS

- Fully Socketed
- Gold Plated Bus Connectors
- Only "B" Series 68XX Components Used
- Only Top Grade Logic Circuits Used
- Industrial Grade Components Throughout

THE MEMORIES

DM-64

- Field Proven
- Proprietary Memory Control Logic
- Fully Transparent Refresh
- Tested at 2.5 MHz Operation

DM-512

- 512K Bytes on a Single S-64 Board
- 16 Bit Power and 8 Bit Compatibility
- Runs in Existing S-50 Systems where Physical Space Allows
- Full 24 Bit Addressing
- Fully Transparent Refresh

THE PRICES

Because of the variety of configurations possible, full pricing cannot be given. Representative prices are:

- **64K 6809 HELIX..... $2495**
- **256K 6809 HELIX..... $2895**
- **512K 6809 HELIX..... $3750**
- **512K 68000 HELIX..... $4195**

HAZELWOOD COMPUTER SYSTEMS

907 E. Terra, O'Fallon, Missouri 63366 (314) 281-1055

Dealer and OEM Inquiries Invited. We support our Dealers.

Optima is a Trademark of Scientific-Atlanta